AD-A215 737

COMPUTER-AIDED DESIGN FOR BUILT-IN-TEST (CADBIT) - Technical Issues

Grumman Aerospace Corporation
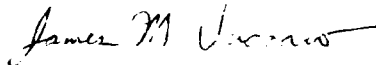
Theo J. Monahos

DTIC
ELECTE
DEC 13 1989
S B D

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

89 12 1

This report has been reviewed by the RADC Public Affairs Division (PA) and is releasable to the National Technical Information Services (NTIS) At NTIS it will be releasable to the general public, including foreign nations.
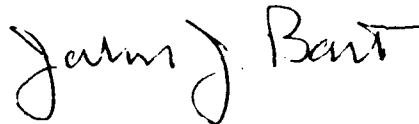
RADC-TR-89-209, Vol I of (three) has been reviewed and is approved for publication.

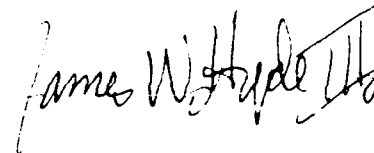APPROVED: *James M Vaccaro*

JAMES M. VACCARO
Project Engineer

APPROVED: *John J Bart*

JOHN J. BART
Technical Director
Directorate of Reliability & Compatibility

FOR THE COMMANDER: *James W Hyde III*

JAMES W. HYDE III
Directorate of Plans & Programs

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS N/A

2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE N/A	

4. PERFORMING ORGANIZATION REPORT NUMBER(S) N/A	5. MONITORING ORGANIZATION REPORT NUMBER(S) RADC-TR-89-209, Vol I (of three)

6a. NAME OF PERFORMING ORGANIZATION Grumman Aerospace Corporation	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Rome Air Development Center (RBES)

6c. ADDRESS (City, State, and ZIP Code) South Oyster Bay Rd Bethpage NY 11714	7b. ADDRESS (City, State, and ZIP Code) Griffiss AFB NY 13441-5700

8a. NAME OF FUNDING / SPONSORING ORGANIZATION Rome Air Development Center	8b. OFFICE SYMBOL (If applicable) RBES	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F30602-86-C-0219

8c. ADDRESS (City, State, and ZIP Code) Griffiss AFB NY 13441-5700	10. SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO. 62702F	PROJECT NO. 2940	TASK NO 01	WORK UNIT ACCESSION NO 01

11. TITLE (Include Security Classification)

COMPUTER-AIDED DESIGN FOR BUILT-IN-TEST (CADBIT) - Technical Issues

12. PERSONAL AUTHOR(S)
Theo J. Monahos

13a. TYPE OF REPORT Final	13b. TIME COVERED FROM Sep 86 TO Sep 88	14. DATE OF REPORT (Year, Month, Day) October 1989	15. PAGE COUNT 168

16. SUPPLEMENTARY NOTATION

N/A

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Built-In-Test BIT data base
12	05		CADBIT Software Printed Circuit Boards
14	02		Computer Aided Design

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Computer Aided Design for Built-In-Test (CADBIT) Final Report consists of three volumes organized as follows. Volume I is a general description including introduction, automated procedure, data base, menus, CAD and BIT survey, and recommendations. Volume II contains a description of the BIT data base library element and BIT library elements for 13 BIT techniques, which were found to be suitable for CADBIT. Volume III contains the CADBIT software requirements specification to be used as a basis for encoding the CADBIT software modules and the creation of its data base.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☑ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL James M. Vaccaro	22b. TELEPHONE (Include Area Code) (315) 330-4205	22c. OFFICE SYMBOL RADC (RBES)

DD Form 1473, JUN 86   Previous editions are obsolete.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

APPENDICES

1.0 SUMMARY

This report is one of three volumes. The Executive Summary describes the entire CAD-BIT effort and the following Volume Summary is a decription of this volume

1.1 EXECUTIVE SUMMARY

CAD-BIT is a development program to specify the implementation of an automated procedure to integrate Built-In-Test (BIT) into the design of Printed Circuit Boards (PCBs) on Computer-Aided Design (CAD) workstations. When fully developed, the CAD-BIT software will be capable of operating on generic workstations meeting various standards. These standards include those for operating system (UNIX), programming language (C), and graphical data ir _rchange Initial Graphics Exchange Specification (IGES).

The purpose of this program was to develop the design of the automated procedure. the associated BIT data base, and a software specification for the CAD-BIT module ready for encoding. No coding of the CAD-BIT Module (CBM) was performed except as necessary to test and verify feasibility. CAD workstations and BIT techniques and their applications were also surveyed to determine standards required for the CAD-BIT module implementation and to establish requirements for and define the structure of the BIT data base.

1.1.1 SCOPE

This report describes the development of the CAD-BIT automated procedure, the associated Data Base of BIT Functions, and software specification developed during this contract. The contents of this report are organized into the three volumes described below.

Volume I Technical Issues

Volume I is a general CAD-BIT description and provides useful information for any type of involvement with CAD-BIT. It begins with an Executive Summary describing the work performed under the CAD-BIT contract. It is followed by a detailed description of the automated procedure. The description contains text, flow diagrams of the procedure operations and its data, sets of menu sequences showing menu options, selections. and resulting operations. Algorithms and formulas are included. The CAD-BIT Data Base and its files are described.

Additional topics in Volume I include Menus, the CAD-BIT Feasibility Demonstration. BIT and CAD workstation surveys and Standards Recommendations, SMART-BIT Applications, and a Automated Procedure Evaluation. The Volume also includes an appendix with a BIT library example for the On-Board ROM BIT Technique.

Volume II   BIT Library

Volume II contains a description of the BIT data base library elements and BIT library
elements for the thirteen BIT techniques listed below. The data in Volume II will be used
to encode CAD-BIT's BIT technique data base during the implementation phase. In addi-
tion, it illustrates the required data for adding new BIT techniques. It also provides useful
data to the future circuit designer / CAD-BIT user on the BIT techniques, their implemen-
tation, and the default circuit components.

* On-Board ROM
* Microprocessor BIT
* Microdiagnostics
* On-Board Integration of VLSI Chips BIT (OBIVCB)
* Built-In Logic Block Observer (BILBO)
* Error Detection and Correction Codes
* Scan
* Digital Wraparound
* Pseudo Random Pattern Generator with Multiple Input Shift Register (PRPG/MISR)
* Comparator
* Voltage Summing
* Redundancy
* Analog Wraparound

Volume III   CAD-BIT Software Specification

Volume III contains the CAD-BIT Software Requirements Specification (SRS). This
SRS establishes the requirements for the Computer Software Configuration Item (CSCI)
identified as Computer-Aided Design for Built-In-Test (CAD-BIT) System. It will be
used during the implementation as the basis for encoding the CAD-BIT software modules
and the creation of its data base.

1.1.2 PURPOSE

The purpose of the CAD-BIT system is to provide an automated procedure to aid the
electronic circuit designer in the selection of BIT techniques, the insertion of the associ-
ated BIT circuitry into the PCB design, and to provide a post design evaluation of the
penalties incurred by the addition of BIT circuitry into the PCB functional design.

1.2 VOLUME I SUMMARY

Volume I provides an introductory description of the CAD-BIT automated procedure, the Data Base of Bit Functions, CAD workstation survey, and a BIT survey and usage analysis. The automated procedure, the Data Base of BIT Functions, BIT Survey and usage analysis and several other relevant topics are treated separately in depth in the following sections.

1.2.1 INTRODUCTION

Volume I is composed of the following sections: Automated Procedure Description, CAD-BIT Data Base, CAD-BIT Environments, CAD-BIT Feasibility Demonstration, BIT Survey, CAD Survey, and Automated Procedure Evaluation. Each topic above is summarized in sections 1.2.2 through 1.2.8 and described in detail in sections 2 through 8, respectively.

1.2.2 AUTOMATED PROCEDURE DESCRIPTION

The features of the CAD-BIT automated procedure are described below and are illustrated in the simplified CAD-BIT PROCEDURE OVERVIEW diagram in Figure 1-1.

PCB FUNCTIONAL DESIGN

A PCB functional design is entered into the CAD system in the form of an electrical schematic or logic diagram. During this and all CAD-BIT phases, all host system functions (CAD and other) are available to the designer who is an electrical circuit design engineer and is familiar with the operation and capabilities of the CAD host system.

BIT TECHNIQUE SELECTION

The Bit Technique Selection process consists of the three steps described below. The selection process may be applied any number of times to different partitions of the circuit design if required to provide full BIT coverage.

The designer generates a user design profile which describes, in terms of attributes, the PCB (or a partition of that PCB) being designed. The design can be partitioned so that CAD-BIT can be applied repeatedly on different portions of the total design. This enables several BIT techniques to be applied to the PCB design and any technique to be used repeatedly. Each separate group of BIT circuitry covering a portion of the functional circuitry is considered a separate BIT Group. As part of the profile, different penalty attribute weights can be applied to each BIT group.

The Suitability Module compares the user design profile with the data base of BIT functions to generate a list of BIT techniques suitable to the design or partition.

- 3 -

The Selection Module calculates estimated penalties for each suitable technique and provides an Estimation Penalty Report and Ranking. This shows, for each suitable technique, the estimated associated BIT penalty for each penalty category and for each technique. The overall ranking is computed using normalized penalty values followed by the application of the designer's weighting factors.

```
┌─────────────────────────────────────────┐
│  ┌───────────────────────────────────┐  │
│  │  ┌─────────────────────────────┐  │  │
│  │  │     INPUT CAD PCB           │  │  │
│  │  │     FUNCTIONAL              │  │  │
│  │  │     DESIGN                  │  │  │
│  │  └─────────────────────────────┘  │  │
│  └───────────────────────────────────┘  │
│                   │                      │
│                   ▼                      │
│         ┌─────────────────────┐          │
│         │     SELECTION       │          │
│         │     PROCESS         │          │
│         └─────────────────────┘          │
│                   │                      │
│                   ▼                      │
│         ┌─────────────────────┐          │
│         │     BIT CIRCUIT     │          │
│         │     INPUT TO        │          │
│         │     CAD SYSTEM      │          │
│         └─────────────────────┘          │
│                   │                      │
│                   ▼                      │
│         ┌─────────────────────┐          │
│         │     EVALUATION      │          │
│         │                     │          │
│         └─────────────────────┘          │
│                                          │
│              Figure 1-1                  │
│                                          │
│      CAD-BIT PROCEDURE OVERVIEW          │
└──────────────────────────────────────────┘
```

Figure 1-1

CAD-BIT PROCEDURE OVERVIEW
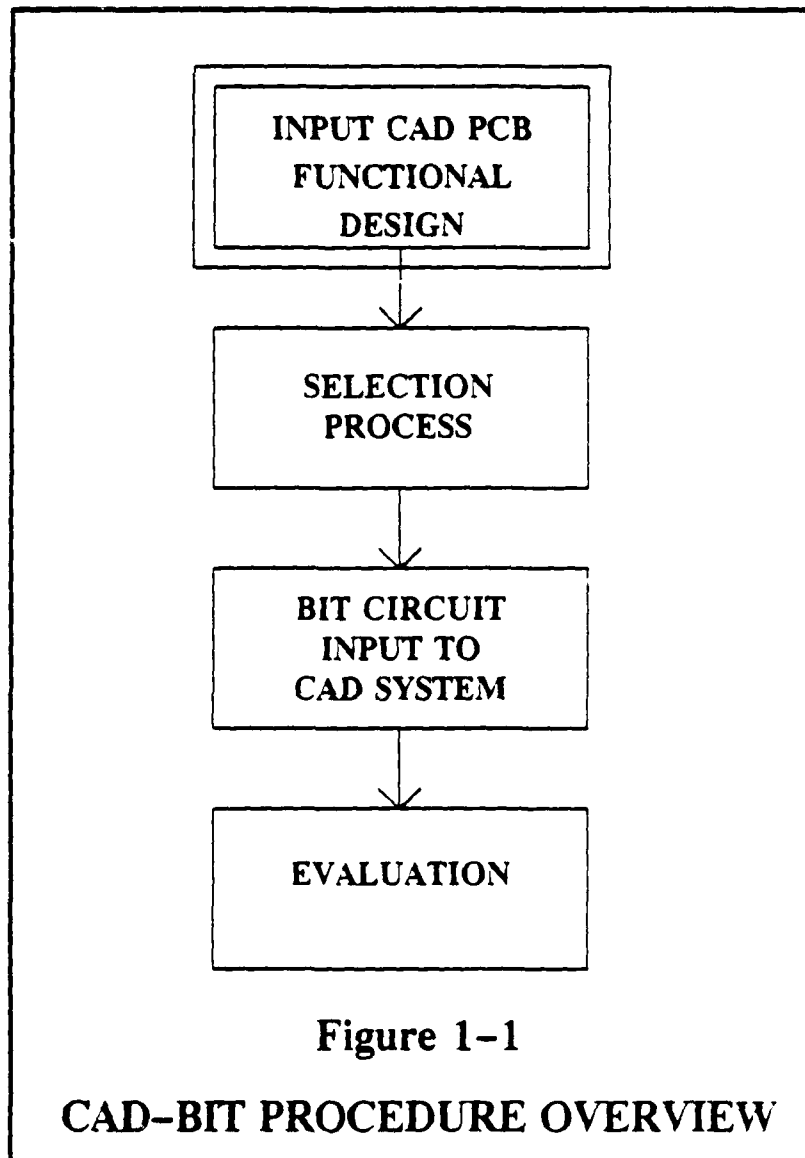
BIT CIRCUIT INSERTION

The designer utilizes functions in the CAD menu to insert BIT circuit elements into the functional design schematic. The CAD menu functions aid the designer to input data used to calculate the number of BIT components required, provide a check-off diagram to keep track of the circuit components inserted, provide tutorial information to aid the interconnection process, execute the CAD component insertion command, insert attributes and attribute values on the BIT circuit elements and provide an estimate of the BIT penalties for each BIT group. The circuitry inserted for any partition is referred to as a BIT group.

EVALUATION PHASE

An evaluation is performed after the insertion of the BIT circuitry to determine BIT penalties (power, area, etc.) incurred due to this insertion. If multiple partitions have been used, each circuit group and its penalties will be shown. A summary of the BIT penalties (added power, weight, etc. due to the BIT circuit elements) is provided.

CAD-BIT ESTIMATION PENALTIES VS. EVALUATION

CAD-BIT provides an estimation of BIT penalties for each suitable technique. This estimation precedes the BIT selection and aids the technique selection process. The evaluation consists of the actual BIT penalties after the BIT circuitry has been inserted into the design.

1.2.3 CAD-BIT DATA BASE

A simplified block diagram of the CAD-BIT data base structure is illustrated in Figure 1-2. The data base is made up of three categories of data files; CAD-BIT System Files, BIT Technique Files, and Design Related Files. The CAD-BIT System Files are those CAD-BIT files which are neither BIT technique related nor design related such as user design profiles. The Bit Technique Files are the data files associated with the individual BIT techniques. These are located within the file structure under the technique name. The Design Related Files consist of the User Design Profiles and other user supplied data pertaining to the PCB designs.

1.2.4 CAD-BIT MENUS

Figure 1-3, CAD / OPERATING SYSTEM ENVIRONMENT MENUS, describes the upper level CAD-BIT menus. Shown are two distinct menus; one for each environment (CAD and Operating System environments). The Operating System Menu in Figure 1-4 contains menu items associated with non-CAD functions. These functions, however, do have a bearing on the CAD menu functions by providing instructions (tutorials) and aid-

ing the BIT circuit insertion and the BIT evaluation processes as described later in this report.

Figure 1-2

CAD-BIT DATABASE OVERVIEW

Figure 1-3
CAD / OPERATING SYSTEM ENVIRONMENTS AND MENUS

Figure 1-4   OPERATING SYSTEM MENU
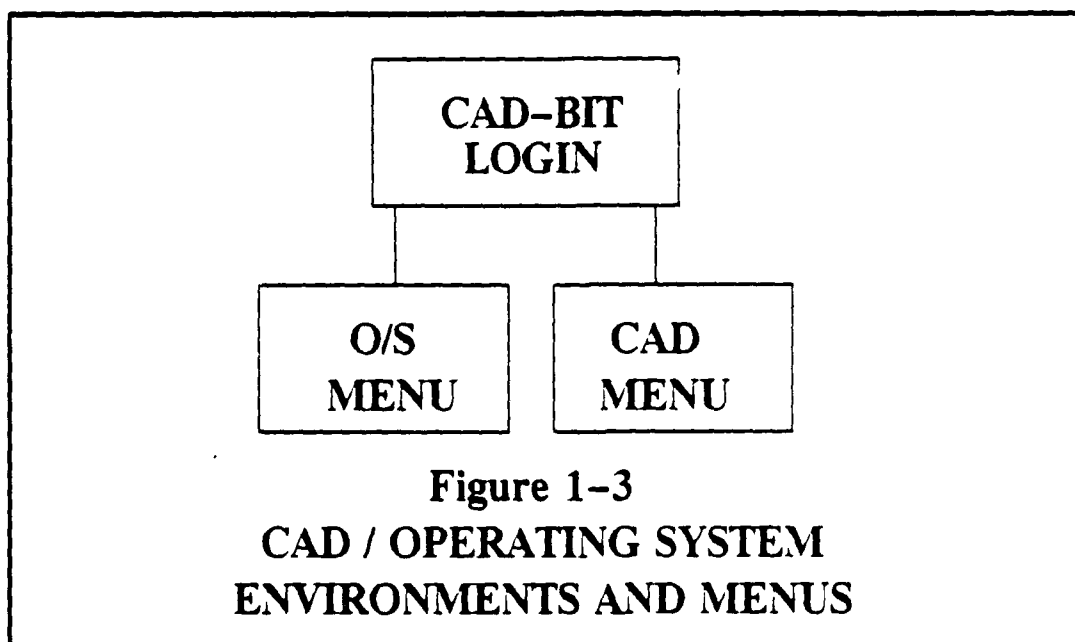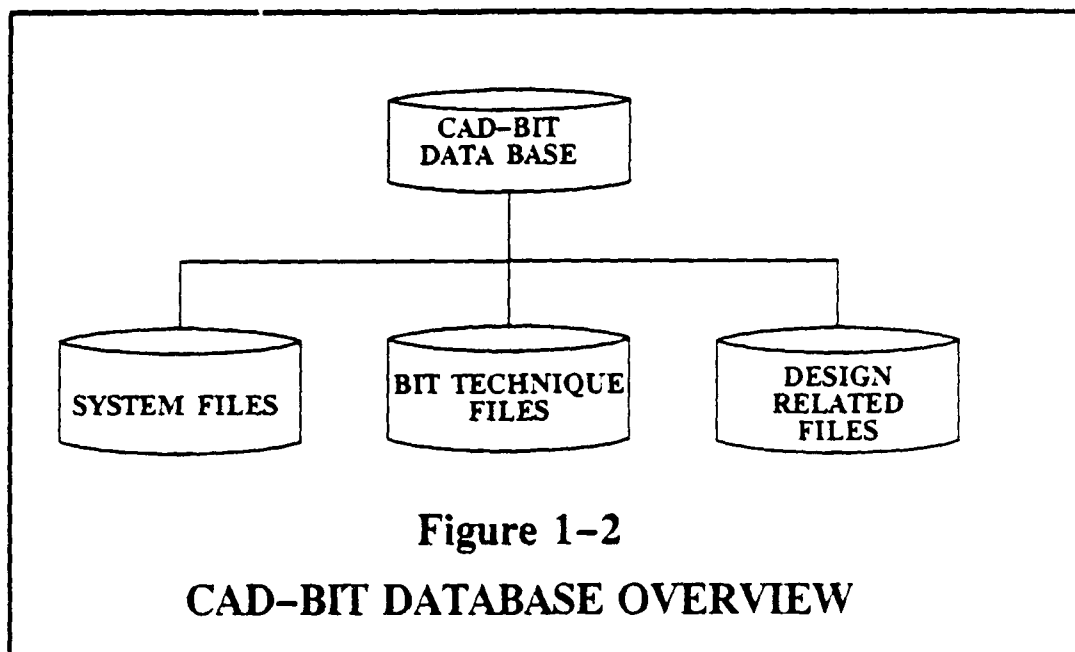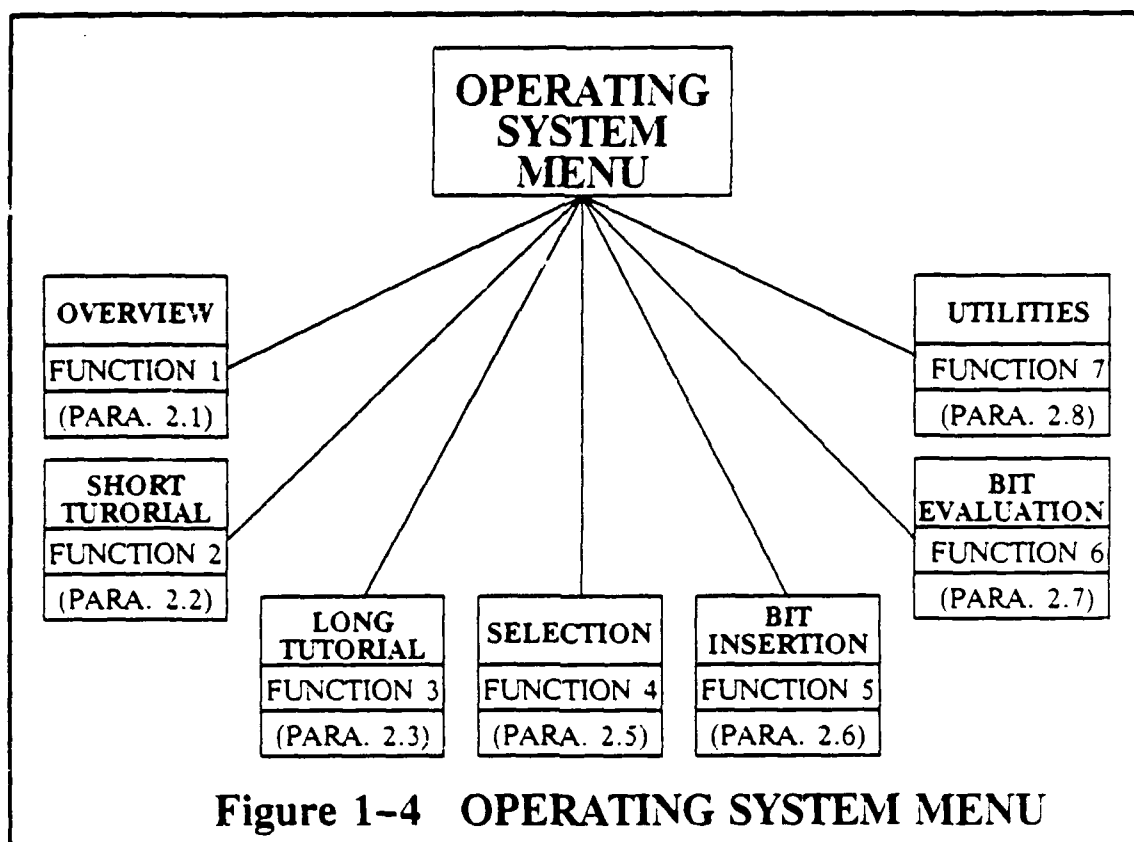
1.2.5 CAD-BIT FEASIBILITY DEMONSTRATION

A CAD-BIT demonstration was prepared to explore the feasibility of CAD-BIT concepts and to develop a user-friendly scenario. The demonstration proved that the concepts described in volumes II and III are sound. The demonstration became a development prototype upon which the proposed CBM and required specification are based. The demonstration also identified the limits of transportability and the areas where system customization is required.

1.2.6 BIT SURVEY

A BIT/PCB design survey was performed as part of this contract. The survey included current designs from different electronic equipment/systems of various types (digital, analog, and hybrid) using a variety of BIT techniques. The survey also included an extensive literature search, data from two committees promoting testability buses, and internal corporate BIT expertise. The survey results were used to determine BIT techniques and parameters relevant to CAD-BIT and to its required data base of BIT design options described in Volume II.

1.2.7 CAD SURVEY AND RECOMMENDATIONS

A CAD Workstation Survey was performed to identify the extent of industry standardization and to determine CAD de-facto standards and trends. The survey information was used to identify and recommend standards for programming languages, graphics data transfer protocols, and operating systems to be used for CAD-BIT.

Figure 1-5 summarizes the CAD survey data. The results of the survey were a factor in establishing the recommendations for the standards to be used in the generation of the CAD-BIT specification and the implementation of the CAD-BIT module during the implementation phase. The following paragraphs summarize the principle recommendations arising from the CAD survey. The data for this survey was gathered between September 1986 and February 1987.

OPERATING SYSTEM RECOMMENDATIONS

The operating system recommended for CAD-BIT is UNIX. The selection of UNIX follows directly from the survey data summarized in Figure 1-5. This recommendation is not based solely on a count of which operating systems were found to be in use by the greatest number of CAD vendors. Much consideration was given to other factors such as trends, support by industry and government agencies, the relative strengths of the various operating systems, and especially the relative merits of the various operating systems in use on CAD systems today and the effect on the use and implementation of the operating system on the CAD-BIT Module (CBM).

UNIX has won acceptance by CAD vendors as the most acceptable standard operating system for CAD systems. It provides the most powerful capabilities and utilities for the high end workstations. These high end workstations are required for the design of the full range of present and future PCBs for complex military electronic devices. In addition, UNIX is becoming accepted as the standard operating system for CAD systems. It is one of the few mandatory requirements found in the Navy's CAD specification.

PROGRAMMING LANGUAGE RECOMMENDATIONS

The programming language recommended for the implementation of CAD-BIT is the "C" language. This is the same language in which most of the UNIX operating system is written. Use of C simplifies the interaction of CAD-BIT software with the operating system and provides convenient access to system services and utility programs.

SYSTEM	DATA BASE			MENUS	LIBRARIES	IGES/ EDIF	HARDWARE
	TYPE	STRUCTURE	FORMAT				
CADNETIX	ASCII	OPEN	OBJECT	ICONS SCREENMENUS	- SUPPLIED - USER DEFINABLE	EDIF	CADNETIX BM PC-AT SUN MICROSYSTEMS AT&T others
CALMA				POP UP MENUS SCREEN MENUS	- SUPPLIED - USER DEFINABLE	EDIF	APOLLO
COMPUTERVISION	ASCII	OPEN	HIERARCHICAL	POP UP MENUS ICONS TABLET MENU (opt)	- SUPPLIED - USER DEFINABLE	EDIF[1]	SUN MICROSYSTEMS
DAISY	BINARY	OPEN	HIERARCHICAL	ICONS POP UP MENUS	- SUPPLIED - USER DEFINABLE	EDIF[2]	DAISY (SCH: IBM-XT with DAISY MONITOR opt)
DATA GENERAL	BINARY	OPEN	RELATIONAL	ICONS POP UP MENUS	- SUPPLIED - USER DEFINABLE	EDIF[1]	DATA GENERAL BM PC
FUTURNET	BINARY	OPEN	HIERARCHICAL	SCREEN MENUS	- SUPPLIED - USER DEFINABLE	EDIF[1]	BM PC-XT TANDY 3000 TANDY 1200 HD
HEWLET-PACKARD[3]					- SUPPLIED - USER DEFINABLE		HEWLET-PACKARD
NTERGRAPH	ASCII	OPEN	HIERARCHICAL	SCH: POP UP MENUS SCREEN MENUS PC: TABLET MENUS SCREEN MENUS	- SUPPLIED - USER DEFINABLE	EDIF[1]	NTERGRAPH WORKSTATION with DEC VAX
MENTOR	BINARY	OPEN	RELATIONAL		- SUPPLIED - USER DEFINABLE	EDIF	APOLLO
ORCATECH	BINARY			SCREEN MENUS	- SUPPLIED - USER DEFINABLE		ORCH 1000 ORCH 3050
RACAL-REDAC	ASCII	OPEN	RELATIONAL	SCREEN MENUS	- SUPPLIED - USER DEFINABLE	EDIF[1]	APOLLO DEC MICRO-VAX
SCIENTIFIC CALCULATIONS	ASCII	OPEN	HIERARCHICAL	POP UP MENUS	- SUPPLIED - USER DEFINABLE	EDIF[1]	DEC MICRO-VAX (SCH: IBM PC opt)
TELESIS[4]				SCREEN MENUS	- SUPPLIED - USER DEFINABLE	EDIF[1]	DEC PDP-11 SUN MICROSYSTEMS others (SCH: BM PC opt)
VALID[4]	ASCII	OPEN		POP UP MENUS	- SUPPLIED - USER DEFINABLE	EDIF	VALID DEC MICRO-VAX (SCH: IBM PC)

NOTES:
1. May not fully conform to the ANSI standard for EDIF at this time.
2. To be included in a future revision.
3. Hewlet-Packard did not respond to a request for information.
4. Valid has announced a proposed merger with Telesis and has begun joint marketing.

Figure 1-5 CAD SURVEY DATA SUMMARY

SYSTEM	OS	LANGUAGE (COMPILERS) (note 6)	GRAPHIC STANDARD	MULTI-PROCESSING	WINDOWS	LINKING	DATA TRANSFER
CADNETIX	UNIX	C (C) (FORTRAN 77)	GKS[7]	YES[8]	YES	NO	● ETHERNET (TCP/IP)
CALMA	UNIX		CORE[7]	YES	YES		● ETHERNET (IEEE 802.3)
COMPUTERVISION	UNIX	FORTRAN 77 (C) (FORTRAN 77) (PASCAL)	GKS[7]	YES	YES	YES	● ETHERNET (TCP/IP)
DAISY	DNIX (UNIX[5]) (SCH: MS-DOS)	C (C) (PASCAL)		YES	YES	YES	● ETHERNET (XEROX INTERNET) (TCP/IP)
DATA GENERAL	UNIX	MAINSAIL (C) (FORTRAN 77) (PASCAL) (MAINSAIL)	GKS[7] GIS[7]	YES	YES	YES	● ETHERNET (IEEE 802.3)
FUTURNET	MS-DOS (CADAT: UNIX)	C & ASSEMBLER	PROPRIETARY	NO	NO	NO	● ETHERNET (TCP/IP)
HEWLET-PACKARD	UNIX						● ETHERNET (TCP/IP)
INTERGRAPH	UNIX	C (C) (FORTRAN) (PASCAL)	GKS[7]	YES[9]	YES[9]	YES[9]	● ETHERNET (IEEE 802.3)
MENTOR	UNIX	C & PASCAL (C) (FORTRAN 77) (PASCAL)	CORE[7]	YES	YES	YES	● DOMAIN
ORCATECH	UNIX	FORTRAN (C) (FORTRAN) (PASCAL)		YES	YES	YES	● ETHERNET RS-232
PACAL-REDAC	UNIX	C & PASCAL (C) (FORTRAN) (PASCAL)	CORE[7] GKS[7]	YES	YES		● ETHERNET STANDARD PROTOCOLS
SCIENTIFIC CALCULATIONS	UNIX	C (C) (FORTRAN) (PASCAL)	GKS[7]	NO	NO	YES	● ETHERNET (TCP/IP) ● RS-232 LAN (4 STATIONS)
TELESIS	UNIX (SCH: MS-DOS)	C & PASCAL (C) (FORTRAN 77) (PASCAL)	GKS[7]				● ETHERNET (TCP/IP) (DECnet)
VALID	UNIX (SCH: MS-DOS)	C & PASCAL (C) (FORTRAN) (PASCAL)	GKS[7]	YES	YES		● ETHERNET (TCP/IP) (DECnet) ● RS-232

NOTES
5 Future revision.
6 Language that software is written in (compilers available on system or as option).
7 Available depending on system.
8 Must submit to another processor via Ethernet.
9 Inter-pro 32 C.

Figure 1-5 (Continued)

DATA INTERCHANGE STANDARD RECOMMENDATIONS

The Initial Graphics Exchange Specification (IGES) is recommended as the graphics data interchange standard for CAD-BIT. The choice of IGES was a difficult one in the following respect. Electrical Computer-Aided Design (ECAD) vendors have indicated that they are supporting the Electronic Design Interchange Format (EDIF), but IGES is being used and is supported strongly by government agencies (refer to paragraph 7.5). Since EDIF is not ready yet, IGES can be expected to emerge as the standard interchange format. If EDIF emerges as the preferable standard when the implementation phase is to begin, there will be no impact on the CAD-BIT specification except to substitute EDIF for IGES.

1.2.8 AUTOMATED PROCEDURE EVALUATION

An evaluation of the proposed automated procedure showed it to be feasible and practical to use. However, limitations in direct portability across CAD systems because CAD vendors' menus are embedded in the CAD application software were noted. This requires customization of the menus for different CAD systems. This procedure is cost-effective to implement, as principle tasks in porting CAD-BIT to other CAD systems are the menu customization cited above and graphics figure translations via IGES. Data base extensibility for both the technique expansion and the parameter expansion are easily handled through the data base design. A detailed evaluation is provided in section 8.

2.0 DETAILED AUTOMATED PROCEDURE DESCRIPTION

This section provides a detailed description of the CAD-BIT automated procedure. In addition, it describes all supporting elements including on-line overview, help functions, tutorials, and utilities. The overview and tutorials are used to introduce a new user to CAD-BIT and aid users in understanding various aspects of the BIT techniques including advantages, disadvantages, and application material. The tutorials aid the user in the understanding and application of CAD-BIT and its techniques.

The section begins with a discussion of the CAD-BIT overview and tutorial menu functions (Operating System Menu Functions 1, 2, and 3 of Figure 2-1). Following that is the the detailed procedure of the CBM including Technique Selection, BIT Circuitry Insertion and BIT Penalty Evaluation (Operating System Menu Functions 4, 5 and 6 of Figure 2-1). Finally a discussion of the CAD-BIT utilities (Operating System Menu Function 7) is given. Menu Function 8, Exit Operating System Menu, simply terminates this menu and will not be discussed further.

In addition to the main CAD-BIT Operating System level menu, a corresponding CAD level menu is also available. CAD-BIT functions are implemented, beginning with the operating system menu, and traversing each menu for that function. Messages are issued when it is required to change menu environments. Although it is possible to implement CAD-BIT without a CAD level menu, the user would need to be more familiar with the CAD-BIT operations. Such an implementation would be far less user-friendly.

2.1 CAD-BIT OVERVIEW

The CAD-BIT OVERVIEW, Operating System Menu Function 1, is a brief one screen description of the overall function of CAD-BIT. It is implemented by printing a short description on the workstation. It is chosen by selecting the Operating System Menu Overview item causing a short description of CAD-BIT to be printed on the screen. The process is illustrated in Figure 2-2. The OVERVIEW is shown in Figure 2-3.
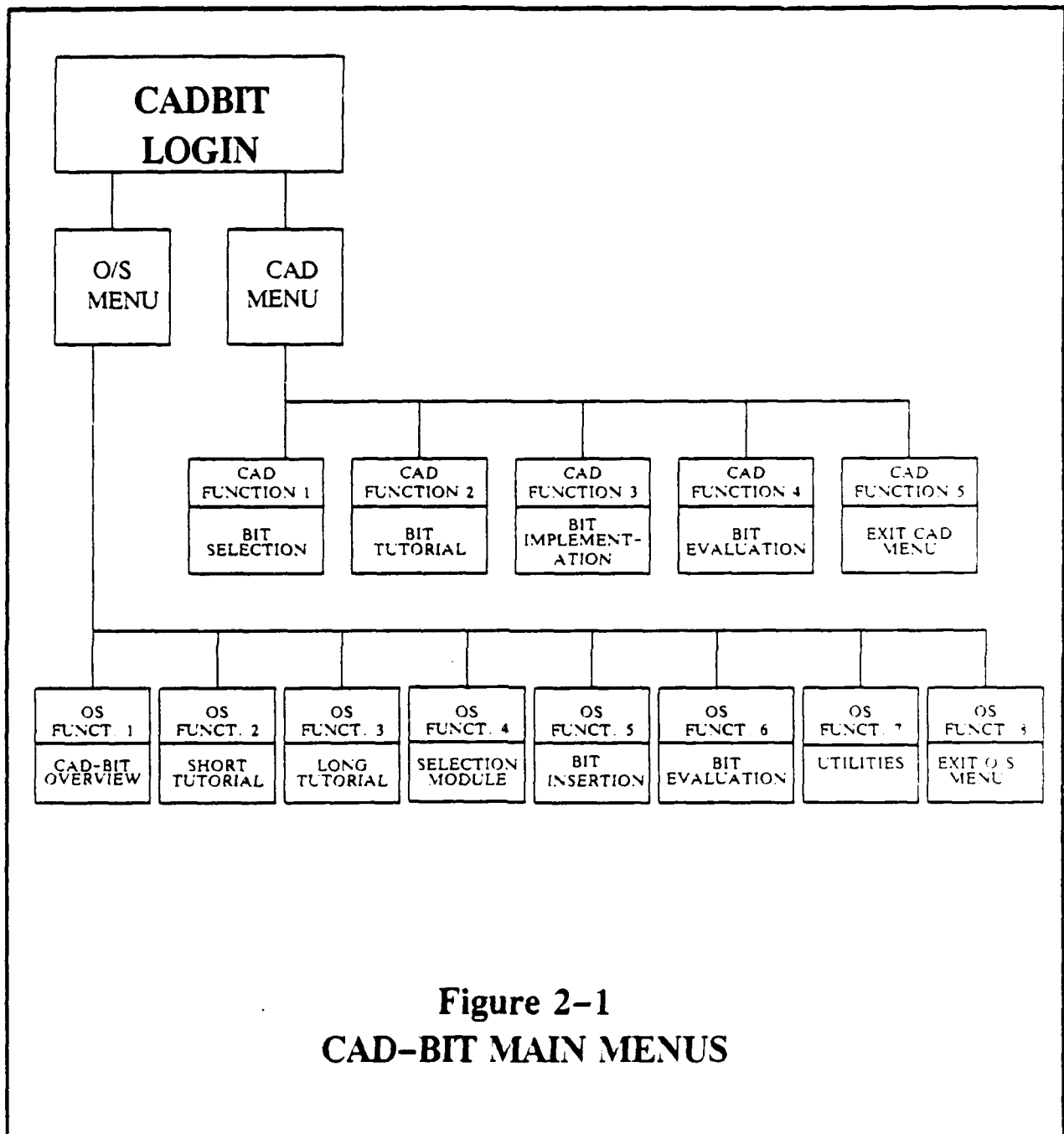
Figure 2-1
CAD-BIT MAIN MENUS

**Figure 2-2
CAD-BIT OVERVIEW**
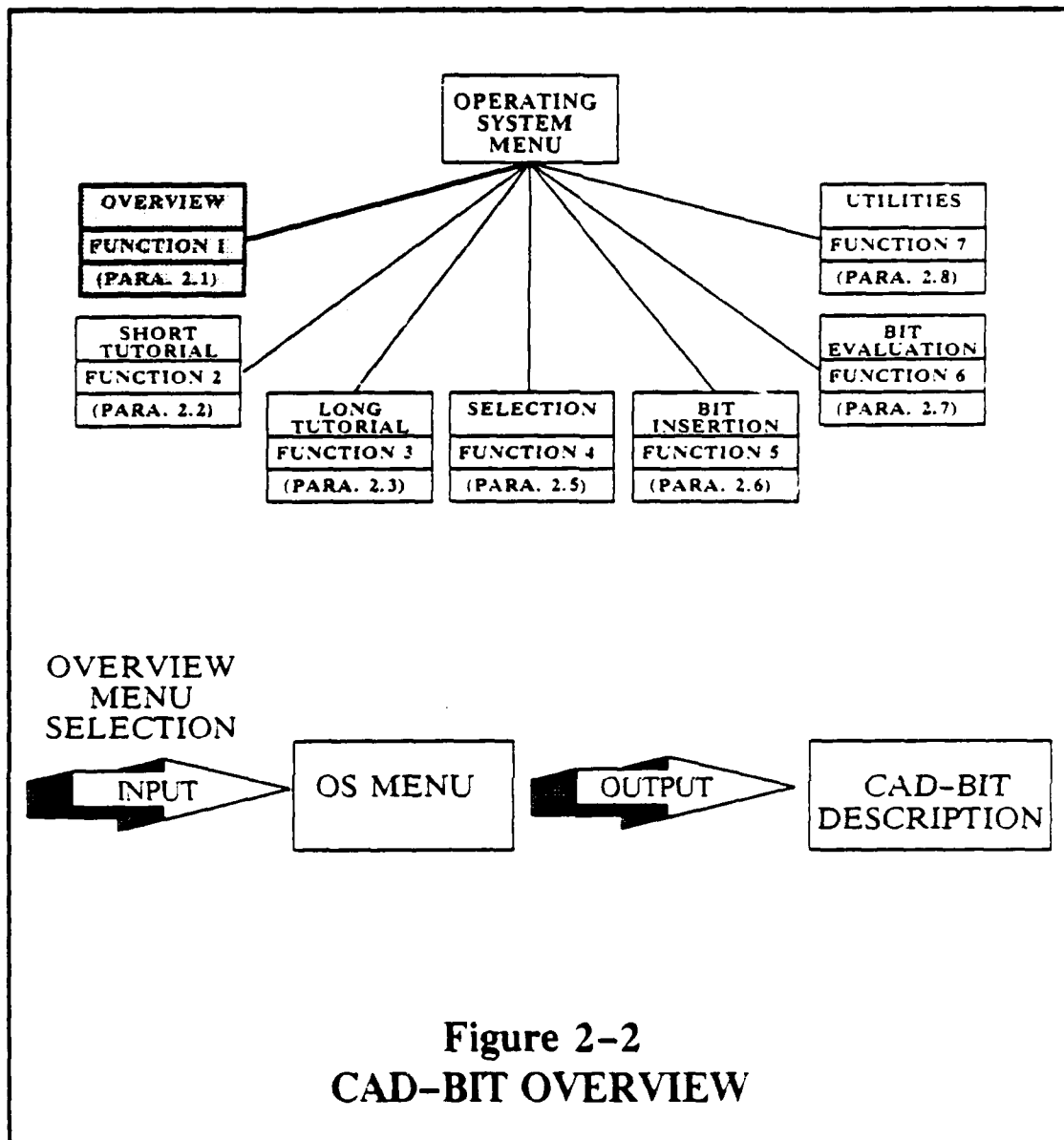
The CAD-BIT module is made up of two different environments. There is a graphics CAD-BIT environment and an operating system CAD-BIT environment. The graphics CAD-BIT activities are controlled through the graphics CAD-BIT environment. The operating system CAD-BIT environment controls all CAD-BIT operations taking place in UNIX. You are currently in the CAD-BIT operating system environment.

The graphics CAD-BIT environment is used to perform all graphics operations. This menu will sometimes refer the user to the operating system menu for any CAD activities occurring in UNIX.

The Selection and Tutorial modules are accessed through the operating system CAD-BIT environment. The BIT Insertion and Evaluation modules are accessed through both the operating system and graphics environments. The menu items in each environment will prompt the user when it is necessary to change environments.

Figure 2-3
CAD-BIT OVERVIEW DESCRIPTION

2.2 SHORT TUTORIAL

The SHORT TUTORIAL, operating system environment Function 2, is a CAD-BIT Operating System Menu selection which is used to provide a narrative description of any BIT Technique. The short tutorial is intended to be a one screen description of each technique. The tutorial may reference any number of graphics figures which help clarify the description. At any time, these figures may be inserted into the CAD environment so that they can be viewed along with the tutorial text. The tutorial text and associated figures form part of the BIT Technique data base. An example of a short tutorial is given in Appendix A. Figure 2-4 outlines the operation of this menu item.

Figure 2-4
SHORT TUTORIAL
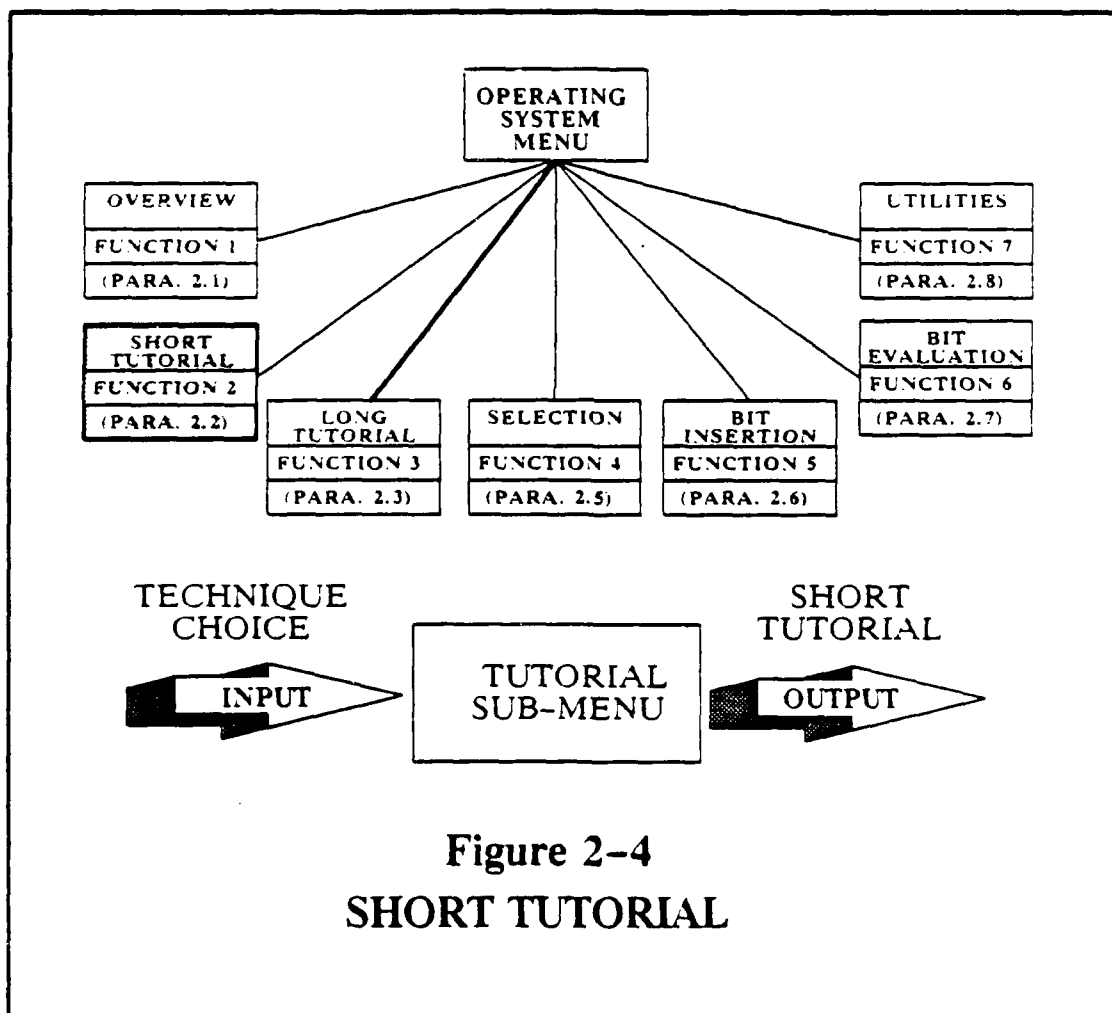
Figures 2-5 through 2-11 provide a detailed sequence of events describing the short tutorial. Included in this sequence is the insertion of a tutorial figure. The lightly shaded areas represent the active environment menu (graphics or operating system). Heavier shading is used to show selected menu items.

The sequence begins with the CAD-BIT operating mode shown in Figure 2-5. The graphical menu is shown on the right entitled "CAD MENU". The CAD-BIT operating system menu is shown in the lower central area and is entitled "CAD-BIT OS MENU". The SHORT TUTORIAL item is selected from the operating system environment menu by choosing menu item number 3. This selection is shown highlighted in the figure.

Upon choosing the short tutorial, a CAD-BIT technique menu is displayed as shown in Figure 2-6. This is generated from the CAD-BIT techniques list to be discussed later in this report. From the figure highlighting, it can be seen that the the ON-BOARD ROM technique is selected. This selection causes the short tutorial text to be displayed in the window as shown in Figure 2-7. It is noted here that the tutorial window size can be changed by the user if required. This is true of other windows as well.

The tutorial makes references to tutorial figures. These figures may be selected by use of the CAD MENU as shown in Figure 2-8, where the CAD-BIT TUTORIAL ("TUTOR" icon) item has been chosen. Note that in this figure the operating system menu is now unshaded and the CAD menu is shaded (selected). For clarity, CAD menu items shown include only a subset of the actual menu selections available. The TUTOR icon selects the TUTOR menu sublevel as shown in Figure 2-9. Here, the techniques are displayed and the "OB-ROM" (On-Board ROM) icon has been selected. The reason for the dual path of tutorial and technique in the CAD / operating system environments is because for a transportable CBM, there is no way to coordinate these choices. The CAD system's menus are generally built into the CAD application software.

In Figure 2-10, the "OB-ROM" tutorial figures choices are displayed and "FIG-1" selected. The selection of this icon generates the CAD command to insert this figure into the CAD system drawing space. The command is displayed in the CAD COMMAND WINDOW (lower left) and the user selects the location for the selected figure to be placed. This is shown by the arrow in Figure 2-10.

Figure 2-11 shows the result. The selected tutorial figure is now displayed in the CAD drawing area while the tutorial text is displayed and read in the CAD-BIT OS MENU window. The tutorial figure may be scaled, moved, or deleted as desired by the user.
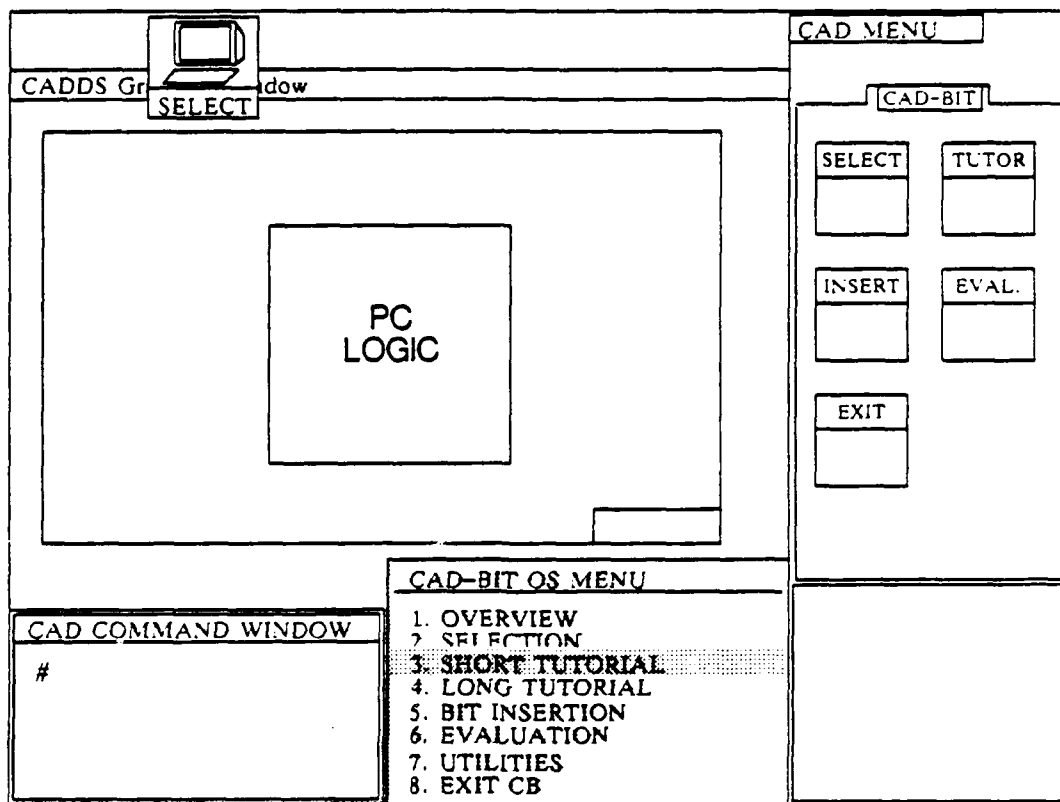
Figure 2-5
CAD-BIT OPERATING MODE: SELECTING
OPERATING SYSTEM ENVIRONMENT SHORT TUTORIAL
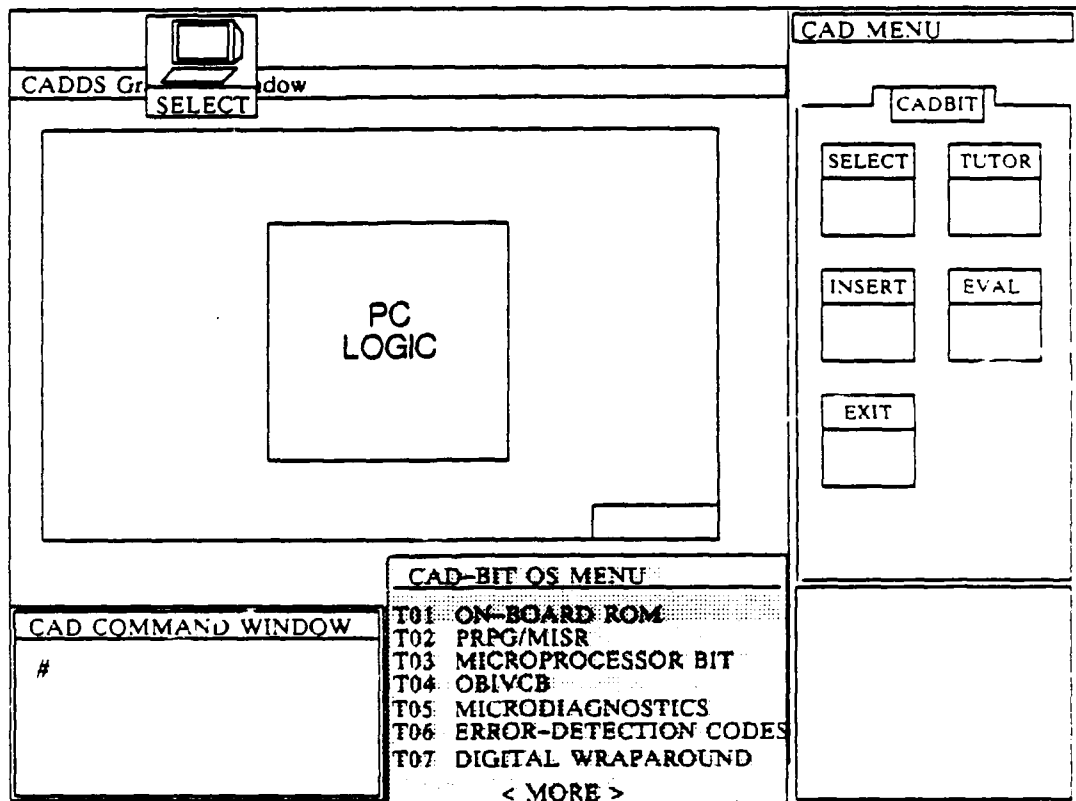
Figure 2-6
SHORT TUTORIAL TECHNIQUE MENU:
SELECTING SHORT TUTORIAL TECHNIQUE
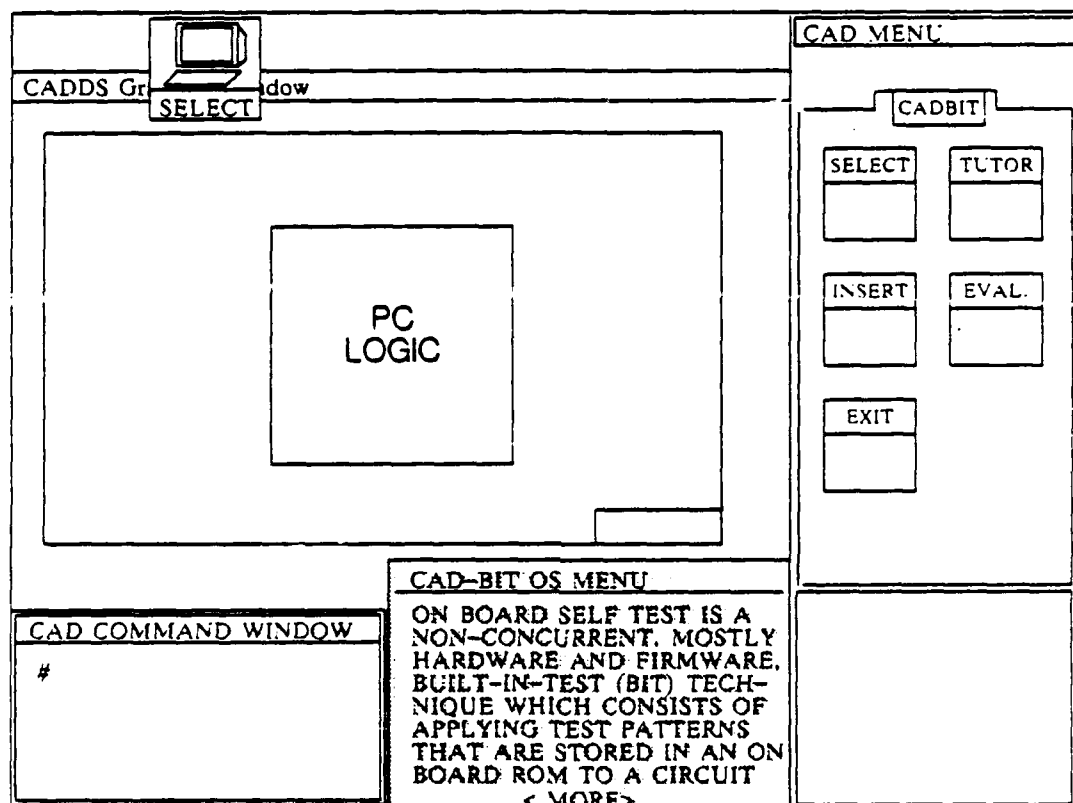
Figure 2-7
SHORT TUTORIAL TECHNIQUE DISPLAY
IN OPERATING SYSTEM ENVIRONMENT
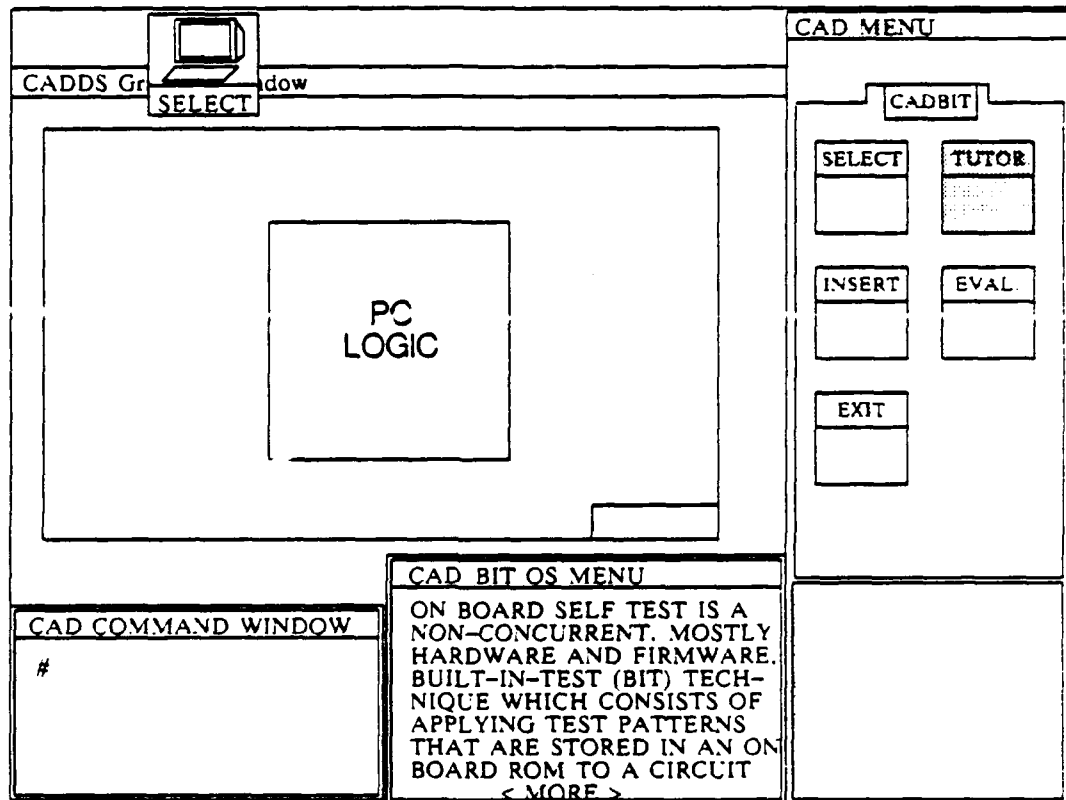
Figure 2-8

**CAD-BIT MENU TUTORIAL SELECTION
IN CAD ENVIRONMENT**
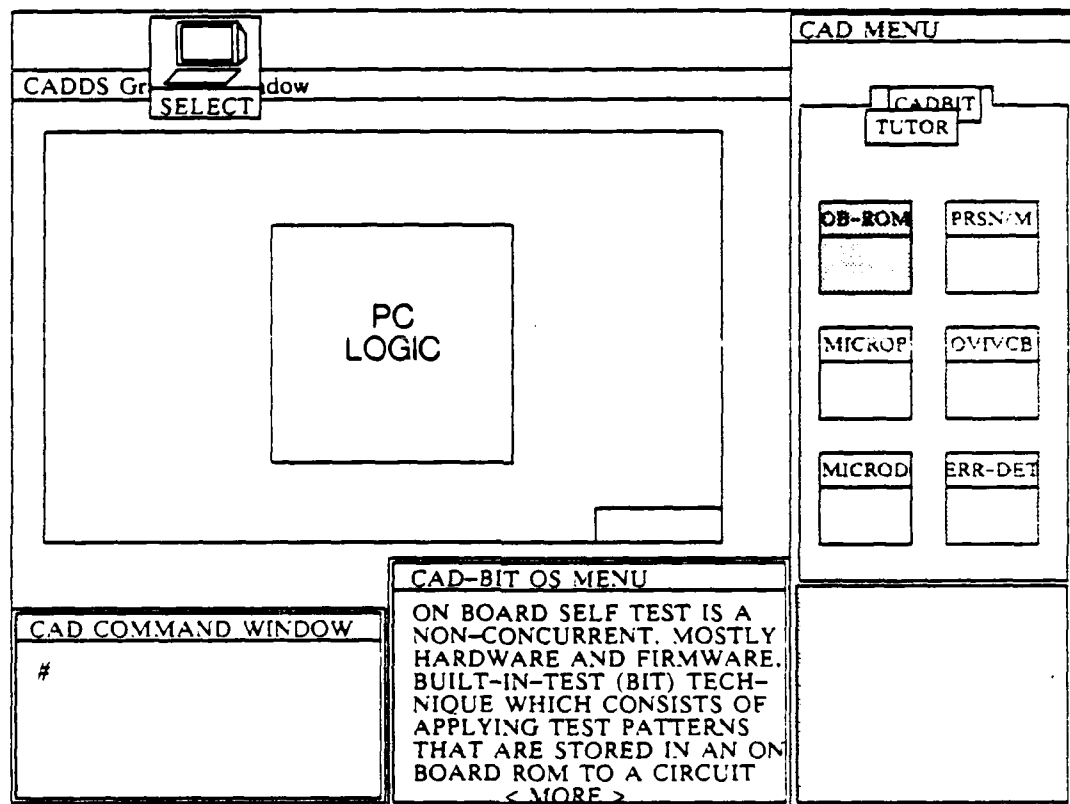
Figure 2-9

TUTORIAL TECHNIQUE SELECTION IN CAD ENVIRONMENT
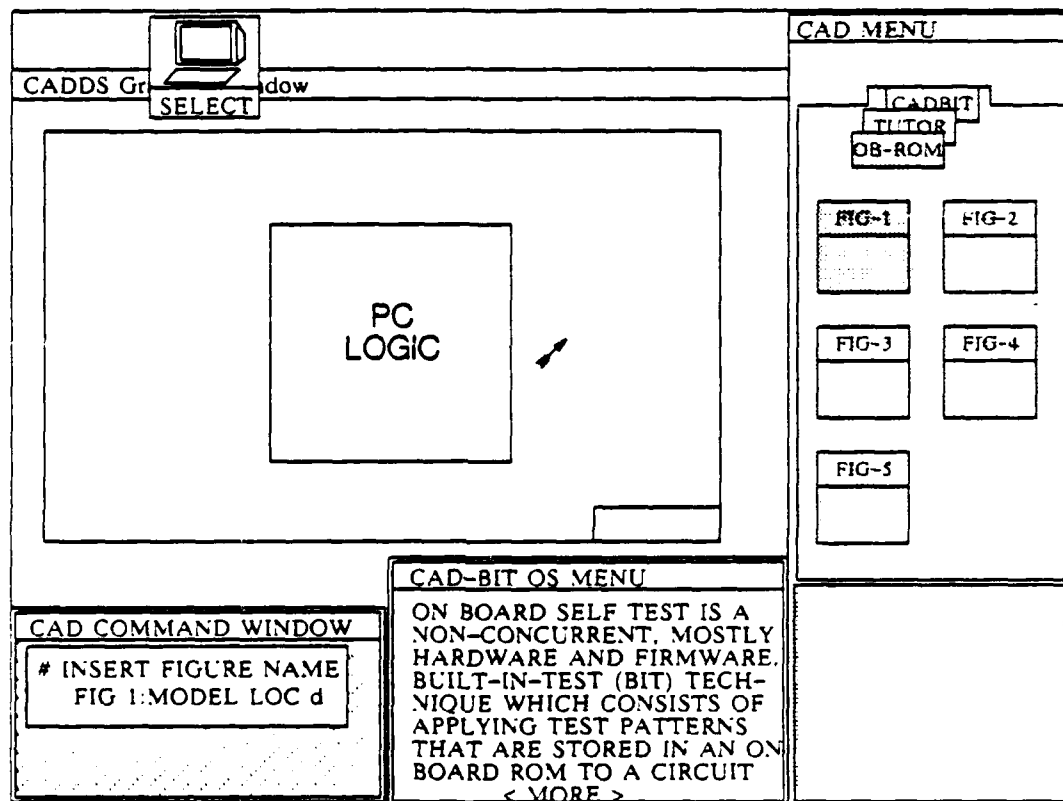
Figure 2-10

TUTORIAL FIGURE SELECTION
IN CAD ENVIRONMENT
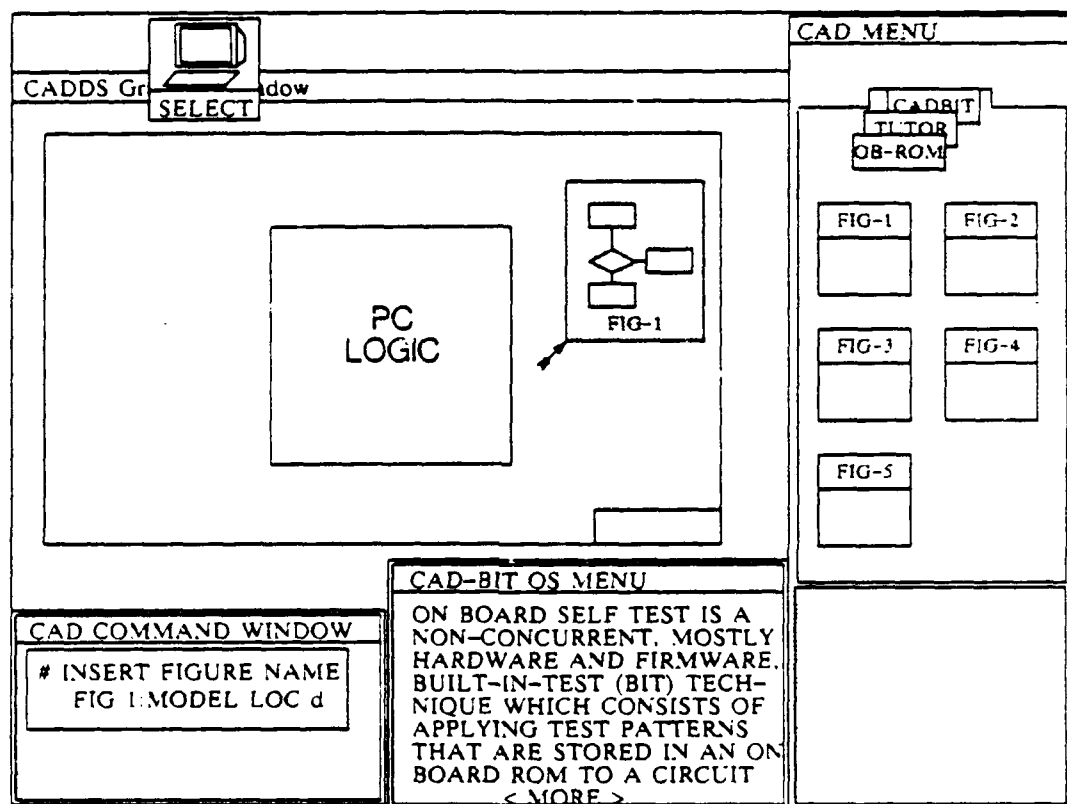
Figure 2-11

**TUTORIAL FIGURE PLACEMENT
IN CAD ENVIRONMENT**

2.3 LONG TUTORIAL

The LONG TUTORIAL, operating system environment Function 3, is a CAD-BIT Operating System Menu item which is used to provide a detailed description of any of the BIT Techniques. The description includes text files containing the following :

A. BIT Sequence Description
B. Advantages
C. Disadvantages
D. BIT Technique Attributes
E. Default Design
F. Part Data Table
G. Bibliography

These text files reference various graphical figures which support the textual descriptions and are viewed by inserting them into the graphical environment in a manner identical to that described in the short tutorial section. The tutorial text and associated figures form part of the BIT Technique data base. An example of a long tutorial is given in Appendix A. Figure 2-12 outlines the operation of this menu item.

Figures 2-13 through 2-15, provide a detailed sequence of events describing the selection of the long tutorial menu item and its options. In Figure 2-13 the long tutorial menu item is chosen from the CAD-BIT operating system environment menu. The technique menu is then displayed and the technique chosen (Figure 2-14). The long tutorial submenu is then displayed and the desired option chosen (Figure 2-15). The selected tutorial text is then displayed and tutorial figures inserted into the CAD window space as required. This process is identical to that for the short tutorial (refer to Figures 2-7 through 2-11).

2.4 PCB FUNCTIONAL DESIGN

PCB Functional Design is referred to as the design of the PCB's functional circuit design (schematic or logic diagram) without any BIT circuitry included. During the functional design and the operation of CAD-BIT, all host system functions are available to the designer. The designer should be familiar with the operation and capabilities of the CAD host system. Since these design functions are provided by the CAD workstation, no CAD-BIT menu items are required.
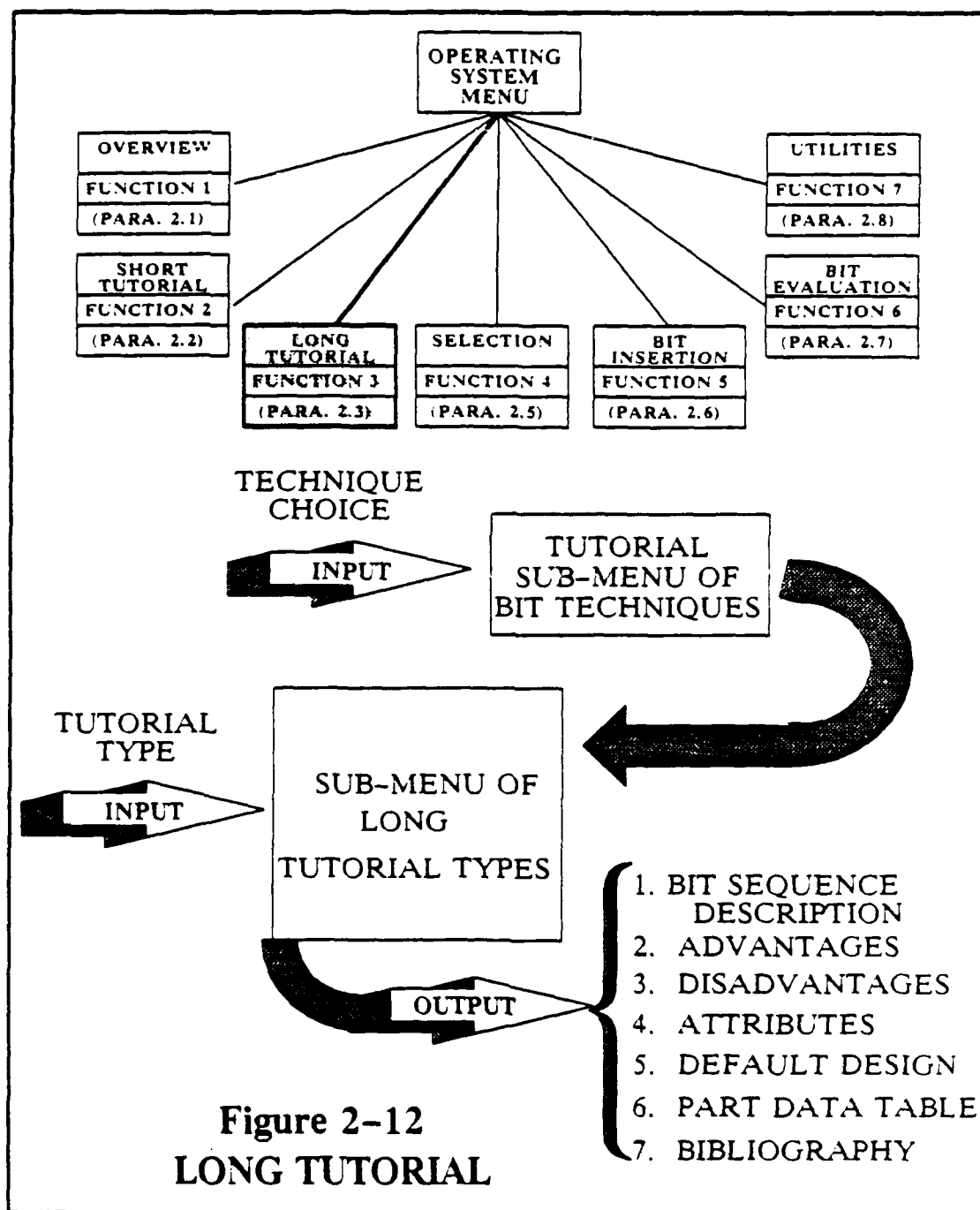
Figure 2-12
LONG TUTORIAL
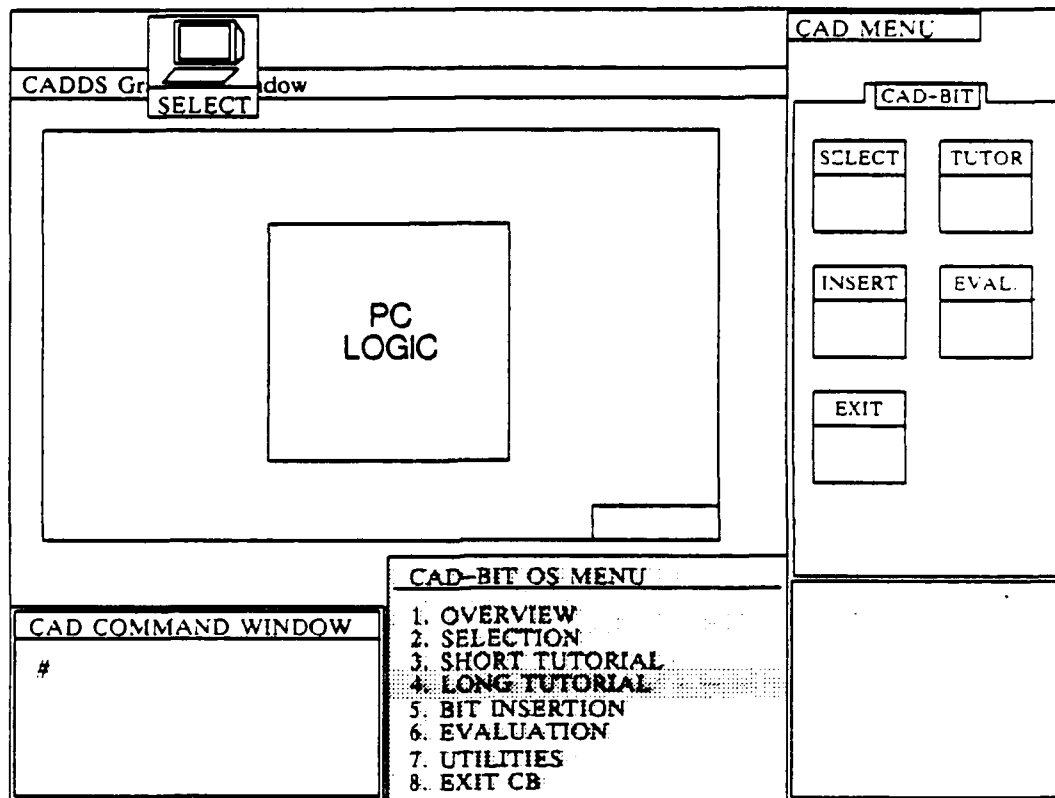
Figure 2–13
CAD–BIT OPERATING MODE: SELECTING
OPERATING SYSTEM ENVIRONMENT LONG TUTORIAL
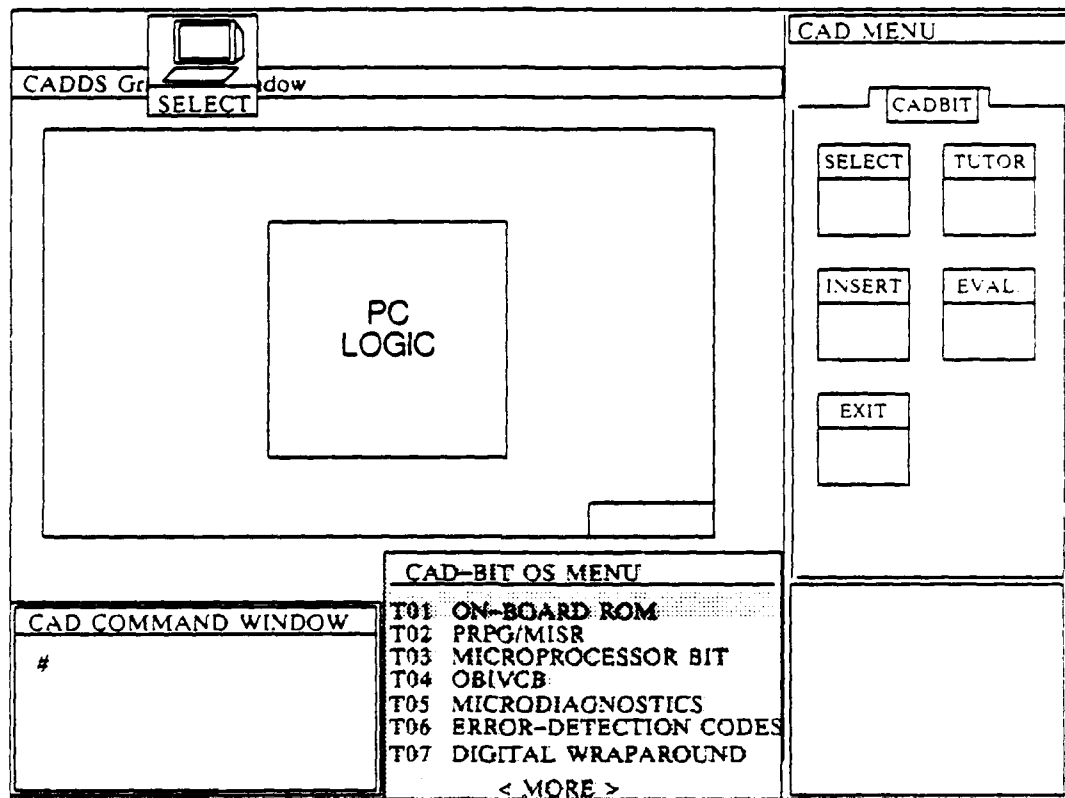
Figure 2-14
LONG TUTORIAL TECHNIQUE MENU:
SELECTING LONG TUTORIAL TECHNIQUE
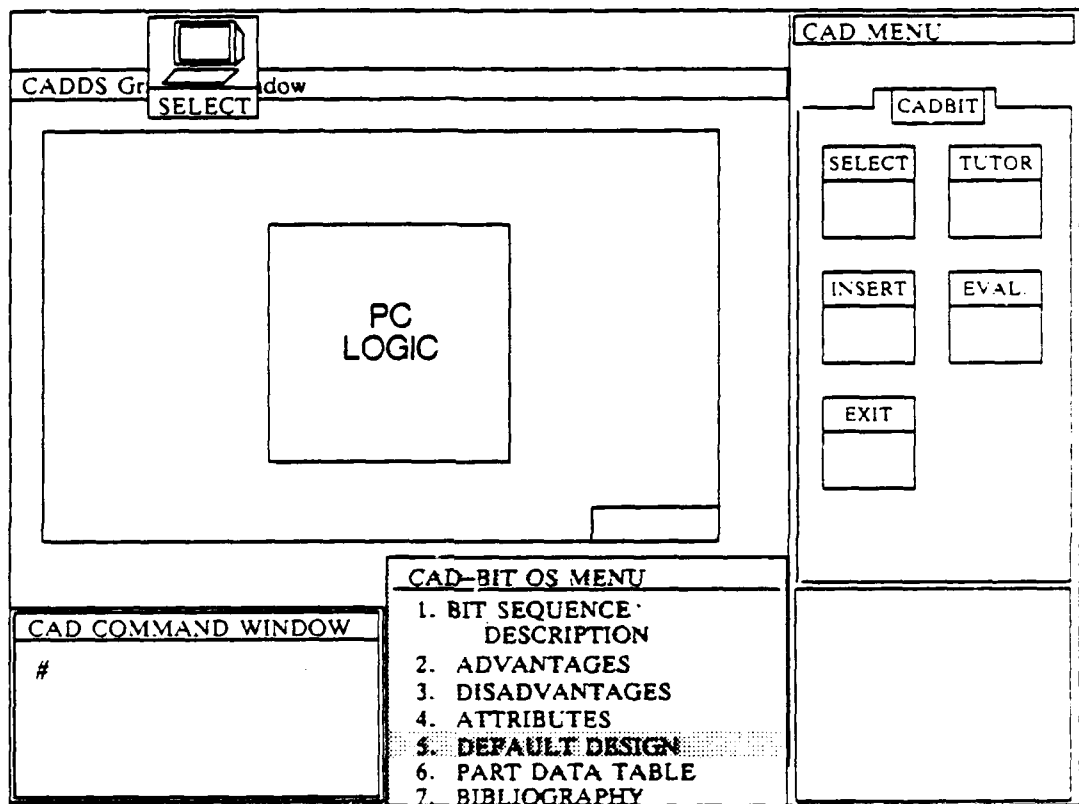
Figure 2–15
LONG TUTORIAL OPTIONS LIST

CAD-BIT has no limiting effects on the CAD system's capabilities at any time. All CAD and Operating System functions are available at all times. Even during the use of CAD-BIT, all standard non-CAD-BIT functions can be used within open windows, or new windows can be opened and used.

As indicated in Figure 2-16, the inclusion of BIT circuitry may result in the PCB functional design being revised to facilitate the inclusion of a BIT technique or to make the design more BIT efficient.

2.5 BIT TECHNIQUE SELECTION

The BIT Technique Selection process, operating system menu Function 4, consists of the three steps described below (see Figure 2-17). The process consists of the generation of a user design profile. The profile is then tested against each BIT technique data base item for suitability. For those techniques which are suitable, a penalty estimation and ranking is calculated resulting in the BIT technique recommendation. The selection process may be applied any number of times to different partitions of the circuit design if required to provide full BIT coverage.

2.5.1 USER DESIGN PROFILE

The designer generates a user design profile which describes, in terms of attributes, the PCB (or a partition of that PCB) being designed. The design can be partitioned so that CAD-BIT can be applied repeatedly on different portions of the total design. This enables several BIT techniques to be applied to the PCB design and any technique to be used repeatedly. Each separate group of BIT circuitry covering a portion of the functional circuitry will be considered a separate BIT group. As part of the profile, different penalty attribute weights can be applied to each BIT group.

PROFILE GENERATION PROCESS

The profile generation process is illustrated in Figure 2-18. The system generates user design profiles by copying either an existing profile or the system default profile into the new profile name. The new profile is then edited under program control where questions are asked and default answers (where applicable) and allowable answers are displayed. The questions are generated by use of a Profile Template File (see Figure 2-19). The answers are stored in the new user Design Profile File, Figure 2-20.

Figure 2-16
IMPLEMENTATION PROCEDURE OVERVIEW

Figure 2-17
SELECTION PROCESS

Figure 2-18
PROFILE GENERATION
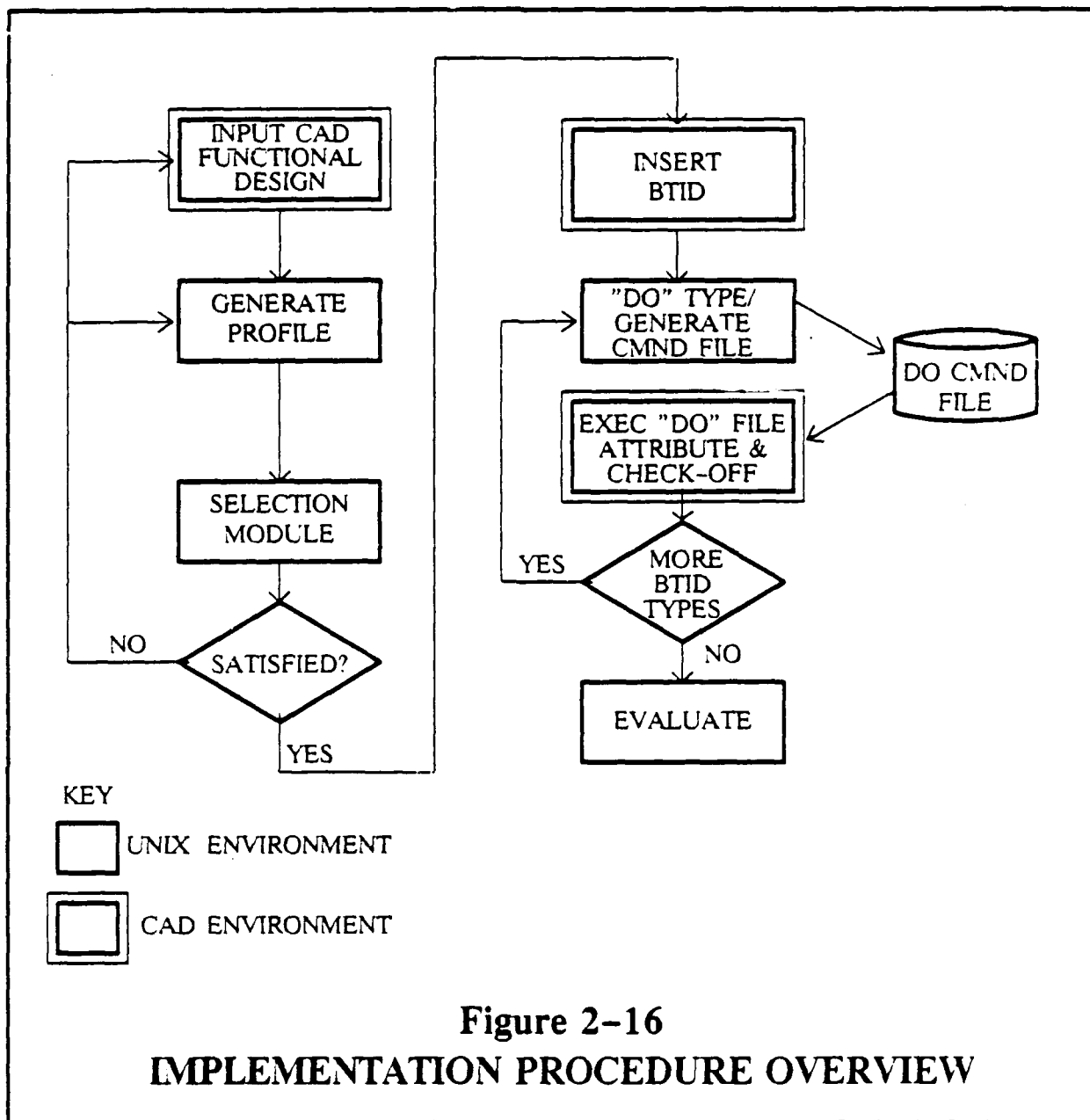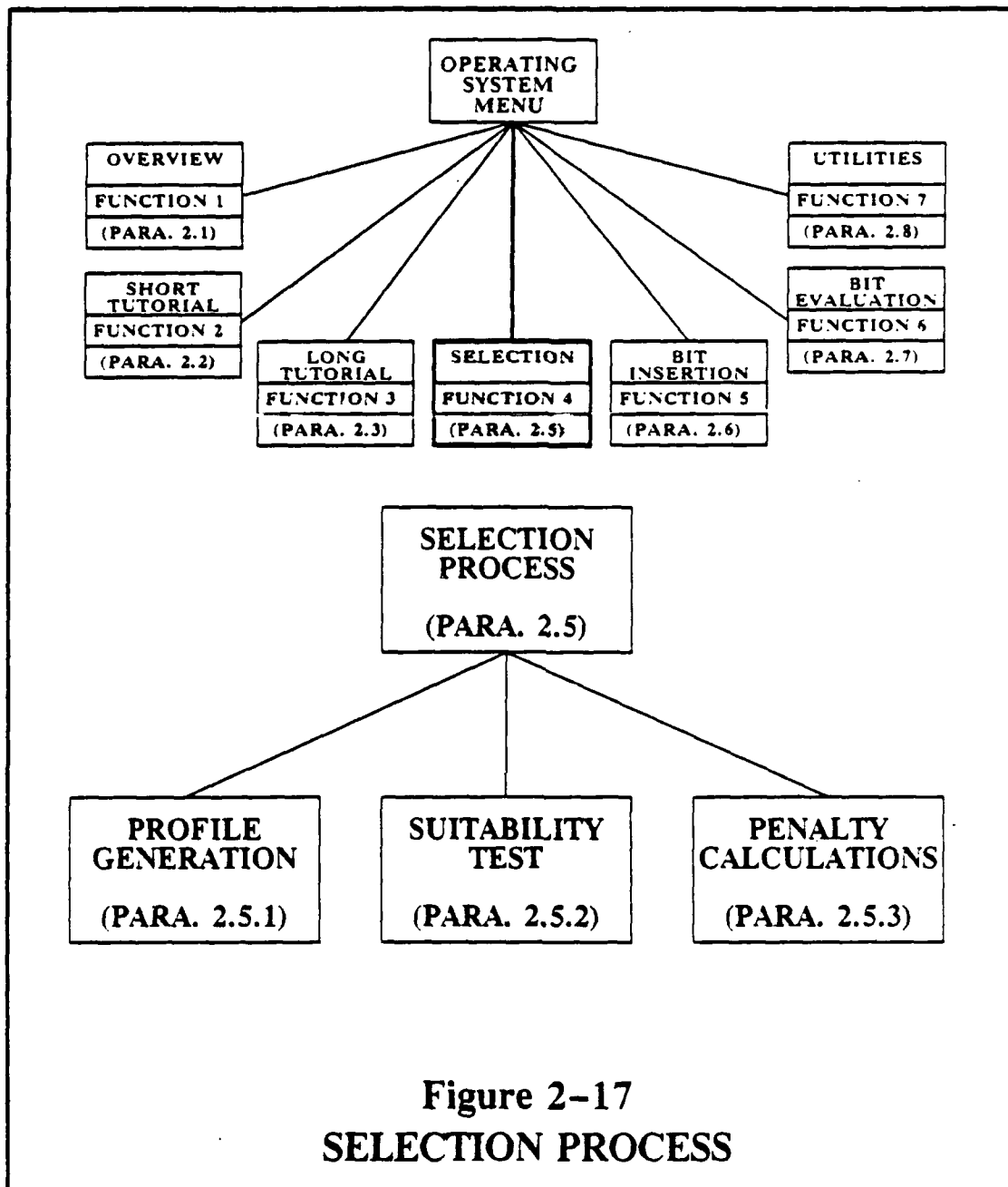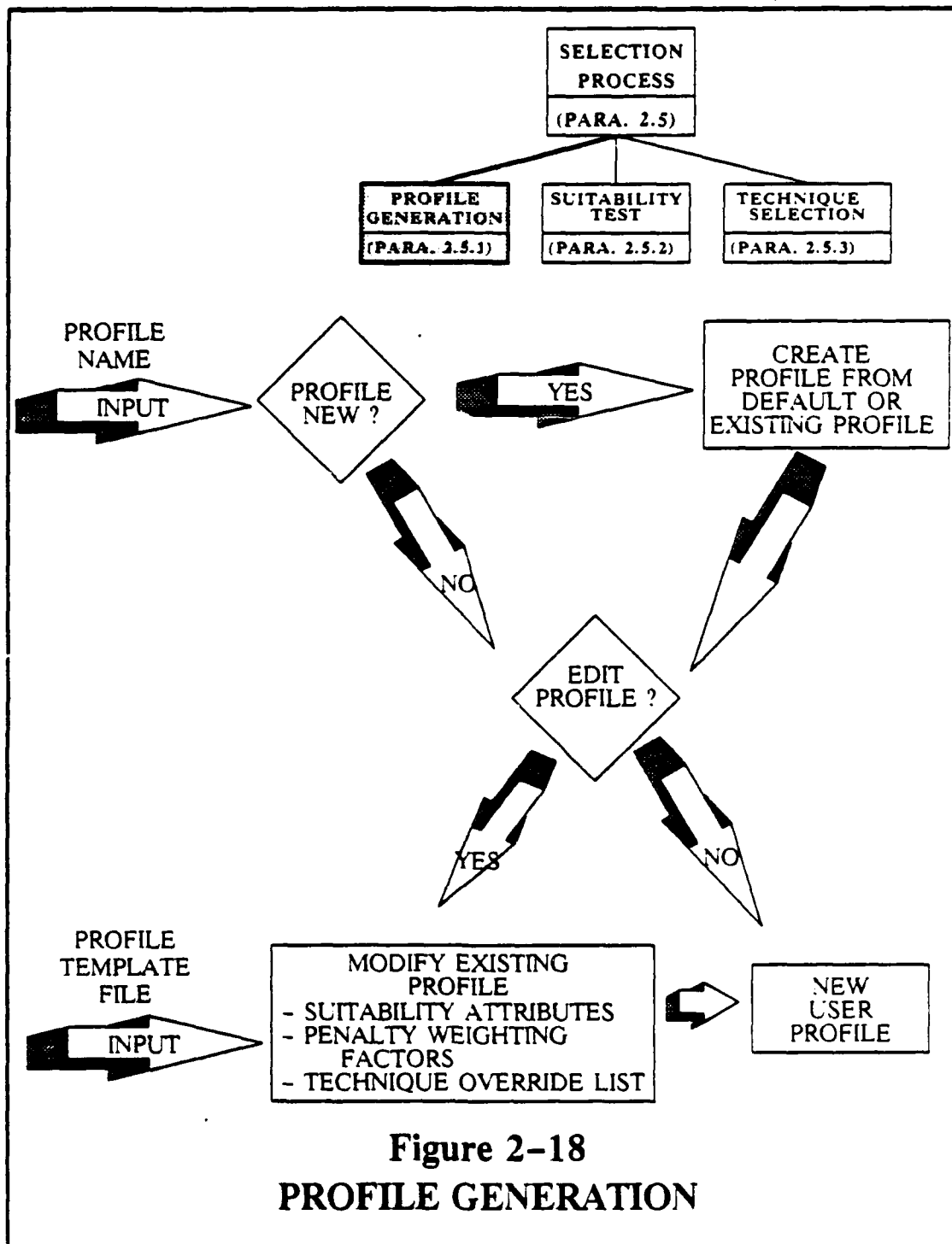
SUITABILITY ATTRIBUTE LIST (SAL)	DEFAULT ANSWER	SUITABILITY ALLOWABLE ANSWERS (SAA)	SUITABILITY QUESTION SET (SQS)
USAGE	D	DAHR	Circuitry Involved?
CONCURRENT	Y	YN	Concurrent testing?
INTERNAL	N	YN	Internal Design?
MICROPROCESSOR	Y	YN	Microprocessor in circuit?

PENALTY PARAMETER LIST (PPL)	DEFAULT PENALTY WEIGHTING FACTORS		PENALTY QUESTION SET (PQS)
AREA	10	#	Area weighting factor?
POWER	10	#	Power weighting factor?
WEIGHT	10	#	Weight factor?
DELAY	10	#	Delay weighting factor?
TESTTIME	10	#	Testtime weighting factor?

TECHNIQUE OVERRIDE LIST (TOL)

EXCLUDE			Forced exclusion?
INCLUDE			Forced inclusion?

KEY:

INDICATES THAT NUMERICAL ANSWER (POSITIVE INTEGER IS REQUIRED).

USAGE ATTRIBUTE QUESTION REQUIRES D (DIGITAL), A (ANALOG), H (HYBRID), or R (RF) ANSWER.

CONCURRENT, INTERNAL, AND MICROPROCESSOR QUESTIONS REQUIRE Y(YES) OR N(NO) ANSWER.

THIS AND ADDITIONAL INFORMATION IS INCLUDED IN THE HELP FILES. SEE FIGURE 2-21.

Figure 2-19
PROFILE TEMPLATE FILE

```
                        SUITABILITY
IDENTIFICATION          ANSWER
SECTION                 SET (SAS)

  USER                  JOHN ENGINEER
  CIRCUIT ID            ABC-123
  DESCRIPTION           SIGNAL PROCESSOR-1

SUITABILITY             SUITABILITY
ATTRIBUTE               ANSWER
LIST (SAL)              SET (SAS)

  USAGE                 D
  CONCURRENT            Y
  INTERNAL              N
  MICROPROCESSOR        Y

PENALTY                 PENALTY
PARAMETER               WEIGHTING
LIST (PPL)              FACTORS

  AREA                  40
  POWER                 10
  WEIGHT                10
  DELAY                 5
  TESTTIME              10

TECHNIQUE
OVERRIDE
LIST (TOL)

  EXCLUDE               T05
  INCLUDE               T01
```

Figure 2-20
SAMPLE PCB USER DESIGN PROFILE
CBM GENERATED

PROFILE TEMPLATE FILE

As seen in Figure 2-19, the contents of the Profile Template File contains three lists: the Suitability Attribute List (SAL), the Penalty Parameter List (PPL), and the Technique Override List (TOL).

The SAL contains the list of attributes used in determining suitability of any technique. Included in the SAL section are the suitability question default answers, the Suitability Allowable Answers (SAA), and the Suitability Question SET (SQS).

The PPL contains the list of penalty parameters for which BIT penalties for any technique will be estimated. Included in the PPL are the Default Penalty Weighting Factor and the penalty weighting questions.

The TOL contains the list of technique override options available. These options, forced inclusions and forced exclusions, allow the user to force particular techniques to be included or excluded from the final list of suitable techniques. This will result in techniques being included or excluded from the final selection penalty estimation and ranking.

The file is an ASCII file and can be easily modified to provide extensibility of suitability attributes and penalty parameters.

USER DESIGN PROFILE CONTENTS

The User Design Profile, Figure 2-20, contains four sections and is similar to the Profile Template File. The User Design Profile contains an identification section, identifying the file owner and the PCB part number and nomenclature, and sections corresponding to the Profile Template File. The Suitability Attribute List contains each suitability attribute and the corresponding attribute value. The Penalty Parameter List contains each penalty parameter and the corresponding penalty weighting factor which applies to this design. The Technique Override List contains the techniques to be forced to be either included or excluded.

PROFILE EDITING

Profile editing proceeds under program control in which each profile entry is displayed with the attribute name and its present value. The user may retain that value by typing the return key, or may enter a new value. The answer is compared to the Suitability Allowable Answers (SAA) to check the answer's validity. This editing applies to each attribute in the Suitability Attribute List (SAL) and each parameter in the Penalty Parameter List (PPL) allowing the suitability attribute answers and the penalty weighting factors to be modified. By typing an exclamation point, all following answers are defaulted. By typing a question mark, the help function is invoked.

PROFILE HELP FUNCTIONS

Profile help functions are available during the profile generation and editing process. These are invoked by typing a question mark (?) in response to any question. A sample help file is shown in Figure 2-21. It explains the question and allowable answers in greater detail and may contain additional information. Since they are added within the CAD-BIT file structure, help function files are extensible within the CAD-BIT Module (CBM) design.

2.5.2 TECHNIQUE SUITABILITY DETERMINATION

The Suitability Module determines which BIT techniques are suitable for the user's design. The process is illustrated in Figure 2-22. Suitability is determined by comparing the User Design Profile's attributes to the BIT Technique Attribute File (Figure 2-24) for each technique in the BIT Technique List (Figure 2-23). Any techniques for which the

---

THE ATTRIBUTE USAGE DETERMINES WHICH OF THE THREE MAIN BRANCHES OF BIT TECHNIQUES WILL BE CONSIDERED.

"D" FOR DIGITAL,
"A" FOR ANALOG,
"H" FOR HYBRID.

QUESTION: IS THE LINE REPLACEABLE MODULE
CIRCUIT UNDER TEST DIGITAL,
ANALOG, OR HYBRID ?

ONLY FIRST CHARACTER IS USED

UPPER OR LOWER CASE IS ACCEPTABLE

Figure 2-21
SAMPLE HELP FILE
FOR PROFILE USAGE QUESTION

attributes are not compatible are eliminated. Techniques in the User Design Profile's forced exclusion list are also eliminated. Techniques in the User Design Profile's forced inclusion list are retained. A suitability list is displayed as in Figure 2-25. In addition, a list of unsuitable techniques is displayed along with an explanation of the suitability attributes which caused the rejection (Figure 2-26).
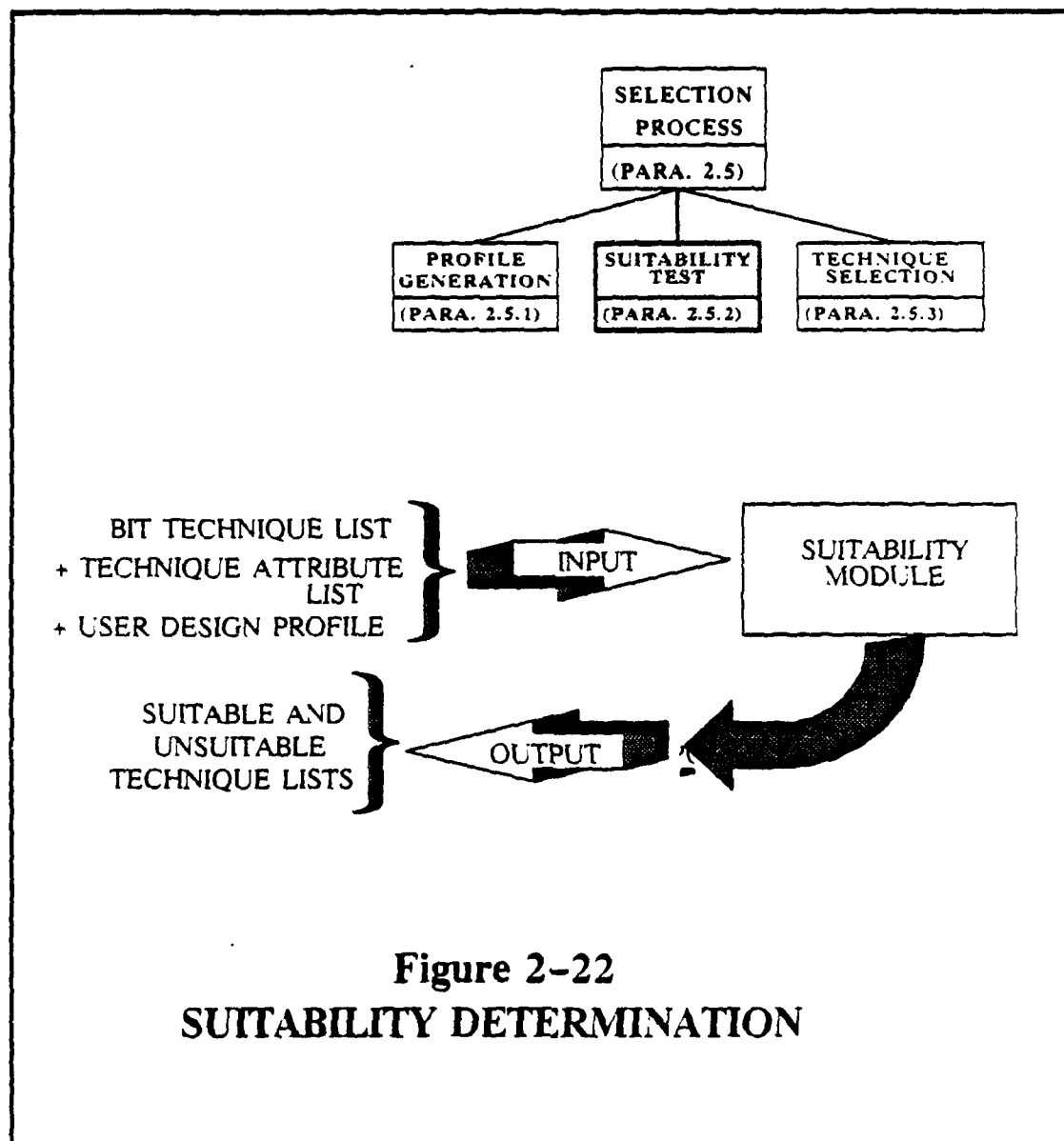
```
                           ┌─────────────┐
                           │  SELECTION  │
                           │   PROCESS   │
                           ├─────────────┤
                           │ (PARA. 2.5) │
                           └─────────────┘
         ┌────────────────────────┼────────────────────────┐
┌─────────────┐          ┌─────────────┐          ┌─────────────┐
│   PROFILE   │          │ SUITABILITY │          │  TECHNIQUE  │
│ GENERATION  │          │    TEST     │          │  SELECTION  │
├─────────────┤          ├─────────────┤          ├─────────────┤
│(PARA. 2.5.1)│          │(PARA. 2.5.2)│          │(PARA. 2.5.3)│
└─────────────┘          └─────────────┘          └─────────────┘
```

BIT TECHNIQUE LIST
+ TECHNIQUE ATTRIBUTE LIST
+ USER DESIGN PROFILE
⟩ INPUT ⟶ SUITABILITY MODULE

SUITABLE AND UNSUITABLE TECHNIQUE LISTS ⟩ OUTPUT

Figure 2-22
SUITABILITY DETERMINATION

T01   ON-BOARD ROM

T02 : PSEUDO RANDOM PATTERN GENERATOR /
        MULTIPLE INPUT SHIFT REGISTER (PRPG/MISR)

T03 : MICROPROCESSOR BIT

T04 : ON-BOARD INTEGRATION OF VLSI CHIP BIT
        (OBIVCB)

T05 : MICRODIAGNOSTICS

T06 : ERROR DETECTION AND CORRECTION CODES

T07 : DIGITAL WRAPAROUND

T08 : BUILT-IN LOGIC BLOCK OBSERVER (BILBO)

T09 : SCAN

T10 : COMPARATOR

T11 : REDUNDANCY

T12 : VOLTAGE SUMMING

T13 : ANALOG WRAPAROUND

Figure 2-23
BIT TECHNIQUE LIST

TECHNIQUE IDENTIFICATION

| TECHNIQUE NUMBER | T01 |
| TECHNIQUE NAME | ON-BOARD ROM |

SUITABILITY ATTRIBUTE LIST (SAL)	REQUIRED ATTRIBUTE VALUE
USAGE	D
CONCURRENT	N
INTERNAL	N
MICROPROCESSOR	N

Figure 2-24

SAMPLE BIT TECHNIQUE ATTRIBUTE FILE

T01 ON-BOARD ROM
T02 PSEUDO RANDOM PATTERN GENERATOR /
     MULTIPLE INPUT SHIFT REGISTER (PRPG/MISR)
T03 MICROPROCESSOR BIT

T08 BUILT-IN LOGIC BLOCK OBSERVER
     (BILBO)

SAMPLE FILE CONTAINING SUITABLE TECHNIQUES.

Figure 2-25
TECHNIQUE SUITABILITY

T03 MICROPROCESSOR BIT REJECTED DUE TO
     ATTRIBUTE USAGE. NO MATCH BETWEEN
     USAGE IN TECHNIQUE (SAA) AND
     USAGE IN PROFILE.

T04 ON-BOARD INTEGRATION OF VLSI
     CHIP BIT (OBIVCB) REJECTED DUE TO
     ATTRIBUTE MICROPROCESSOR. NO
     MATCH BETWEEN MICROPROCESSOR
     IN TECHNIQUE (SAA) AND
     MICROPROCESSOR IN PROFILE.

SAMPLE FILE CONTAINING UNSUITABLE
TECHNIQUES AND A BRIEF EXPLANATION
AS TO WHY A TECHNIQUE WAS REJECTED
DURING THE SUITABILITY TEST PHASE.

Figure 2-26
TECHNIQUE REJECTION

2.5.3 TECHNIQUE SELECTION

The Selection Module, Figure 2-27, calculates estimated penalties for each suitable technique and provides an Estimation Penalty and Ranking report. This report shows, for each suitable technique, the estimated associated BIT penalty for each penalty parameter and a summary penalty ranking for each technique. The overall ranking is computed using normalized penalty values followed by the application of the penalty weighting factors defined by the user. The detailed penalty calculations and ranking procedure is a five step process.

Figure 2-27
PENALTY ESTIMATION AND RANKING
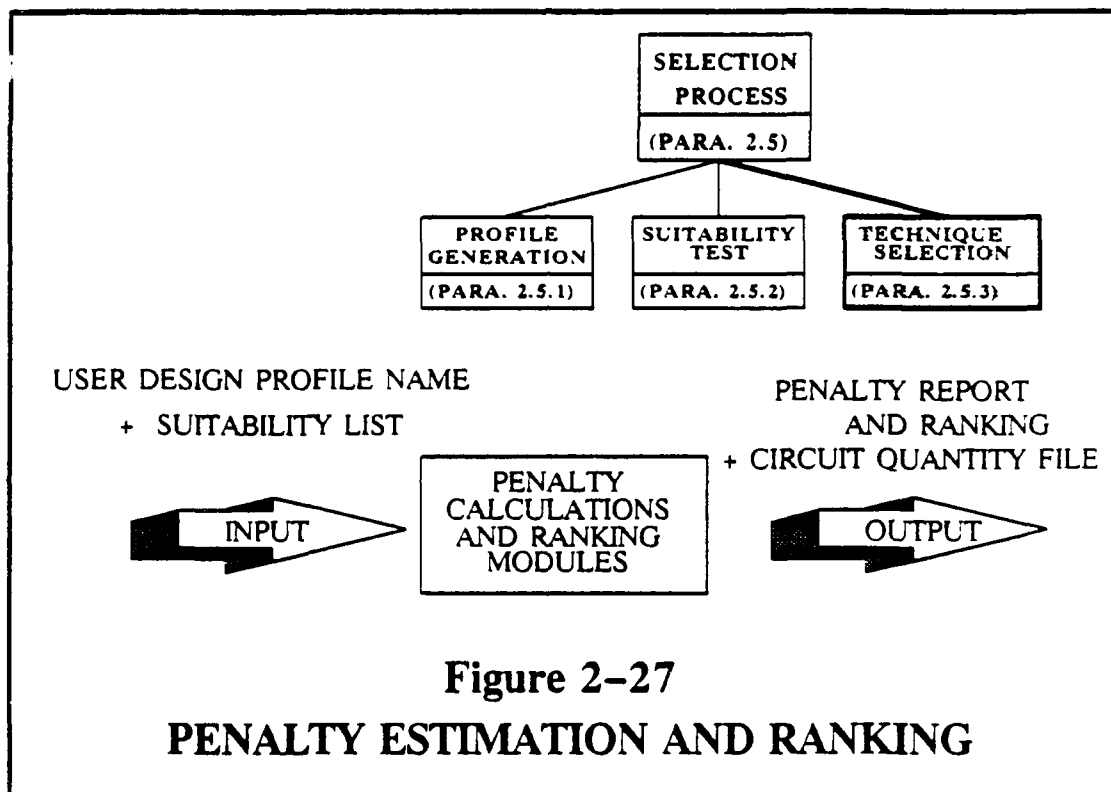
(1)   Determine User Requested Data (URD) Parameter Values

User Requested Data Parameters are those parameters which are necessary to calculate the quantity of each circuit type required to implement a BIT technique or to calculate its BIT penalties. The parameter values are furnished by the CAD-BIT user. The parameters are functions of the characteristics of the PCB design. For example, the URD required for

the On-Board ROM technique are listed below. The parameter variable names (v1, v2, etc.) are included in parentheses.

A. Quantity of primary input pins used by the PCB's operational circuitry (v1).
B. Quantity of primary output pins (v2).
C. Quantity of test patterns to be stored in the ROMs (v3).
D. Test pattern application rate (v4).
E. Estimated initialization rate (v5).

The URD are specific to a particular BIT technique. The number of URD parameters is generally a small number. The URD are obtained during the selection process by asking a corresponding set of questions. This data is saved in a User Related Data File to be reused at the user's option in case the Selection module is rerun.

(2)   Calculate Component Quantities

For each technique circuit type, (ROM, Multiplexer, etc.), a Component Determination Equation (CDE) is used to calculate the number of components required for each circuit type. For example, for the On Board ROM technique the quantity (n) of 8 Bit ROMs with a depth of 2048 bytes is given by the CDE below:

$$n= (v1/8) (v3/2048) + (v2/8) (v3/2048)$$

The parameter n is usually a function of several URD elements and often simply a constant.

(3)   Determine Penalty Parameters

At this point, the technique penalty parameters are calculated. For each suitable BIT technique and for each penalty parameter, a BIT penalty is calculated. The penalty parameters are calculated using Technique Penalty Equations (TPEs). Each TPE is a function of the number of components required and constants for each circuit type. For example, for the On-Board ROM Technique, the area penalty is given by the TPE below:

$$Area= 1.15 ((.375\ nrom) + (.87\ nmux) + (.375\ ncomp) + (.243\ ncntr) +$$

$$(.375\ ndecd) + (.375\ ndel) + (.243\ njkff) + (.243\ nctrlg)) \text{ square inches}$$

where   nrom = the number of ROMs,

nmux = the number of multiplexers,

ncomp = the number of comparators,

ncntr = the number of counters,

ndecd = the number of decoders,

ndel = the number of delay chips,

njkff = the number of jk-flip-flops,

nctrlg = the number of control gates.

Constants in the equations are known physical dimensions and electrical characteristics of the default design. The User Requested Data, Component Determination Equations, and Technique Penalty Equations are defined for each technique in the Volume II, Data Base of BIT Functions. A sample is included in Appendix A.

The penalties are developed into an array, parameter j of technique i is P_{ij}, and the set of p's for all suitable techniques and all penalty parameters constitutes the penalty array. For example, if the On-Board ROM technique is technique number 1 and area is penalty parameter number 1, then the area calculated in the formula above is assigned to the array element P_{11}. The penalty array is shown in step (4) below.

(4)   Calculate Normalized Weighted Penalties

A. The Penalty Array

The penalty array, P_{11}, for k penalty parameters and m suitable techniques is shown below:

P_{11}	P_{12}	$\cdots$	P_{1k}	technique 1
P_{21}	P_{22}	$\cdots$	P_{2k}	technique 2
.			.	
.			.	
.			.	
P_{m1}	P_{m2}	$\cdots$	P_{mk}	technique m

The array has the structure shown in Figure 2-28 where the penalties for each parameter and for each technique are calculated as described above.

BIT TECHNIQUES	BIT PENALTIES			
	AREA	POWER	WEIGHT	$\cdot\cdot$
Technique 1	P_{11}	P_{12}	$\cdots$	P_{1k}
Technique 2	P_{21}	P_{22}	$\cdots$	P_{2k}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
Technique m	P_{m1}	P_{m2}	$\cdots$	P_{mk}

Figure 2-28
PENALTY PARAMETER STRUCTURE

B. Compute the Average Penalty

Compute the average penalty P_j for each penalty parameter j for all m suitable techniques as follows:

$$P_j = \frac{1}{m} \sum_{i=1}^{i=m} P_{ij}$$

C. Normalize the P_{ij} Array

Use the P_j's to normalize the P_{ij} array, resulting in the N_{ij} array where

$$N_{11} = \frac{P_{11}}{P_1} \qquad \text{or} \qquad N_{ij} = \frac{P_{ij}}{P_j}$$

The resulting N matrix is shown below.

$$
\begin{array}{llll}
N_{11} & N_{12} & \cdots & N_{1k} \quad \text{technique 1} \\
N_{21} & N_{22} & \cdots & N_{2k} \quad \text{technique 2} \\
\vdots & & & \vdots \\
N_{m1} & N_{m2} & \cdots & N_{mk} \quad \text{technique m}
\end{array}
$$

D. Apply the Penalty Weighting Factors

For each penalty parameter (area, power, etc.), there also exists a corresponding weighting factor w. These are user supplied during the User Profile Generation or default values are used. The weighting factors make up a single dimension weighting factor, w. Applying w to the N array as follows

$$W_{ij} = w_j \cdot N_{ij}$$

creates a normalized weighted array, W, with the array elements defined as follows:

$$
\begin{matrix}
W_{11} & W_{12} & \cdots & W_{1k} & \text{technique 1} \\
W_{21} & W_{22} & \cdots & W_{2k} & \text{technique 2} \\
\cdot & & & \cdot & \\
\cdot & & & \cdot & \\
\cdot & & & \cdot & \\
W_{m1} & W_{m2} & \cdots & W_{mk} & \text{technique m}
\end{matrix}
$$

(5)   Determine BIT Technique Ranking

R is the column Ranking Array. The selection ranking R_i for technique i is then

$$R_i = \sum_{j=1}^{j=k} W_{ij} \cdot$$

Technique ranking will cause those techniques with a lower overall penalty ranking to be sorted to the top. It is expected that techniques with similar overall penalty values may not clearly define the better ranking as superior. In these cases, engineering judgment and a closer inspection of penalty weighting factors may help discriminate between the closely ranked techniques. Figure 2-29 illustrates the Penalty Estimation and Ranking.

PENALTY PARAMETER	PEN. WEIGHT. FACTORS
AREA	40
POWER	10
WEIGHT	10
DELAY	5
TESTTIME	10

PART OF USER DESIGN PROFILE

T01 : ON-BOARD ROM
T02 : PRPG/MISR
T03 : MICROPROCESSOR BIT
T08 : BILBO

SUITABILITY LIST

CIRCUIT	QUANTITY
ROM	4
MUX	1
⋮	⋮
COMPARATOR	1

CIRCUIT QUANTITY FILE

VIA PENALTY ESTIMATION FORMULAS

ID	TECHNIQUE DESCRIPTION	AREA (mm²)	POWER (mw)	WEIGHT (grams)	TEST TIME (usec)	RANKING
T01	ON-BOARD ROM	4.35	5010	46	195	44
T02	PRPG/MISR	4.82	5825	51	1820	68
T08	BILBO	7.74	4612	32	1650	71
T03	MICROPROCESSOR BIT	2.91	3720	35	2113	84

PENALTY REPORT AND RANKING

Figure 2-29
BIT PENALTY ESTIMATION EXAMPLE
ON-BOARD ROM TECHNIQUE

2.6 BIT CIRCUIT INSERTION

The designer uses the BIT insertion functions in the CAD and operating system environments to insert the BIT circuit elements into the functional design schematic. The CAD menu functions aid the designer in helping to input a BIT Technique Insertion Diagram (BTID) by providing a process guide and check-off diagram to keep track of those circuit components inserted. The BIT insertion functions are also used to execute the component insertion commands, display tutorial information to aid the interconnection process, and to attribute the BIT circuit elements to set-up the post-insertion evaluation of the BIT penalties. The BIT insertion operating system environment, function 5, is used to calculate the required number of each type of BIT components inserted and to generate the CAD component insertion commands. Figure 2-30 illustrates the three step BIT circuit insertion process.

2.6.1 BIT TECHNIQUE IMPLEMENTATION DIAGRAM

The BTID is used as a guide in inserting the BIT circuit components into the PCB functional design. The BTID is illustrated functionally in Figure 2-31. The functional design is shaded and the BIT circuit functions to be added are unshaded. Figure 2-32 illustrates the BTID for the On-Board ROM Technique. The BTID may be inserted into the functional design graphics area temporarily via the CAD environment. As the circuit elements are inserted into the functional design, the corresponding box in the BTID can be checked-off by whatever graphics technique is available in the CAD system. This includes cross-hatching, changing the color of the box, or any other convenient method. When all boxes have been checked-off the BIT technique circuitry has been inserted into the PCB design. Complex BTIDs may be nested if required.

During the BIT insertion, the user has the option of displaying tutorial figures which explain the application of the technique and the interconnection process. The tutorial figures may be inserted into or deleted from the drawing space when required.

The BTID is inserted via the CAD environment menu as indicated in Figures 2-33 through 2-36. The CAD Menu Insertion Selection (Figure 2-33), Insertion Technique Selection (Figure 2-34), and BTID Insertion Selection (Figure 2-35), are sequentially selected. The Insertion Selection and Insertion Technique Selection menu items simply traverse the menu structure. The BTID Insertion Selection item issues the CAD command to insert the BTID. The CAD command is shown in the CAD COMMAND WINDOW of Figure 2-35. Figure 2-36 shows the inserted BTID.

```
                      ┌──────────────┐
                      │  OPERATING   │
                      │    SYSTEM    │
                      │     MENU     │
                      └──────────────┘
┌──────────────┐                                          ┌──────────────┐
│   OVERVIEW   │                                          │  UTILITIES   │
├──────────────┤                                          ├──────────────┤
│  FUNCTION 1  │                                          │  FUNCTION 7  │
├──────────────┤                                          ├──────────────┤
│  (PARA. 2.1) │                                          │  (PARA. 2.8) │
└──────────────┘                                          └──────────────┘
┌──────────────┐                                          ┌──────────────┐
│    SHORT     │                                          │     BIT      │
│   TUTORIAL   │                                          │  EVALUATION  │
├──────────────┤  ┌──────────┐ ┌──────────┐ ┌──────────┐  ├──────────────┤
│  FUNCTION 3  │  │   LONG   │ │SELECTION │ │   BIT    │  │  FUNCTION 6  │
├──────────────┤  │ TUTORIAL │ │          │ │INSERTION │  ├──────────────┤
│  (PARA. 2.2) │  ├──────────┤ ├──────────┤ ├──────────┤  │  (PARA. 2.7) │
└──────────────┘  │FUNCTION 4│ │FUNCTION 2│ │FUNCTION 5│  └──────────────┘
                  ├──────────┤ ├──────────┤ ├──────────┤
                  │(PARA. 2.3)│ │(PARA. 2.5)│ │(PARA. 2.6)│
                  └──────────┘ └──────────┘ └──────────┘
```

OS/CAD ENV.

```
                ┌──────────────┐
                │     BIT      │
                │  INSERTION   │
                │  (PARA. 2.6) │
                └──────────────┘
CAD ENV.                OS ENV.               CAD ENV.
┌──────────────┐  ┌──────────────┐      ┌──────────────┐
│    BTID      │  │  COMPONENT   │      │    "DO"      │
│  INSERTION   │  │  INSERTION   │─────▶│  Commands    │
│ (PARA. 2.6.1)│  │  PROGRAMS    │      │ (PARA. 2.6.3)│
│              │  │ (PARA 2.6.2) │      │              │
└──────────────┘  └──────────────┘      └──────────────┘
```

- BTIDS GUIDE INSERTION PROCESS
- COMPONENT INSERTION PROGRAMS DETERMINE QUANTITIES
  AND CAD FIGURES (COMPONENTS) TO BE INSERTED BY
  GENERATING "DO" COMMANDS IN HOST CAD SYSTEM SYNTAX.
- DO COMMAND ISSUES CAD SYSTEM COMMAND.
- EVENTS OCCUR IN MIXED (CAD AND OS) ENVIRONMENTS.

Figure 2–30
BIT CIRCUIT INSERTION

PCB I/O CONNECTORS

BIT CIRCUIT ELEMENT #1

BIT CIRCUIT ELEMENT #2

BIT CIRCUIT ELEMENT #n

FUNCTIONAL DESIGN

Each BTID is inserted prior to the insertion of the actual BIT circuitry. The BTID's act as guides for the user to follow. They are referenced through the O/S menu BIT Insertion prior to generating the commands for BIT insertion.

Figure 2-31

BIT TECHNIQUE INSERTION DIAGRAM (BTID)

Figure 2–32

ON-BOARD ROM BTID

Figure 2–33

**CAD–BIT MENU INSERTION SELECTION
IN CAD ENVIRONMENT**

Figure 2-34

**INSERTION TECHNIQUE SELECTION
IN CAD ENVIRONMENT**

Figure 2-35

**BTID INSERTION SELECTION
IN CAD ENVIRONMENT**
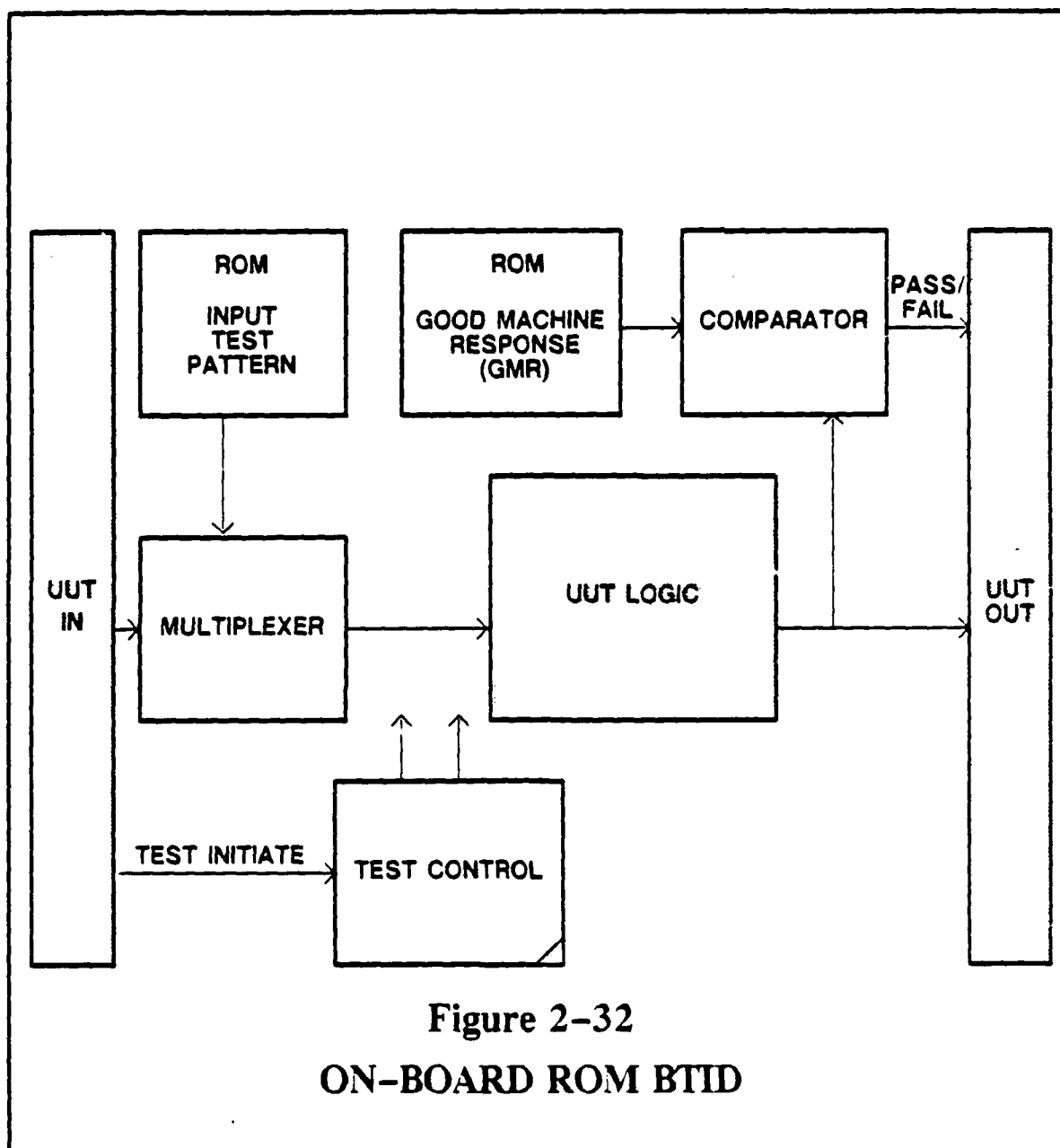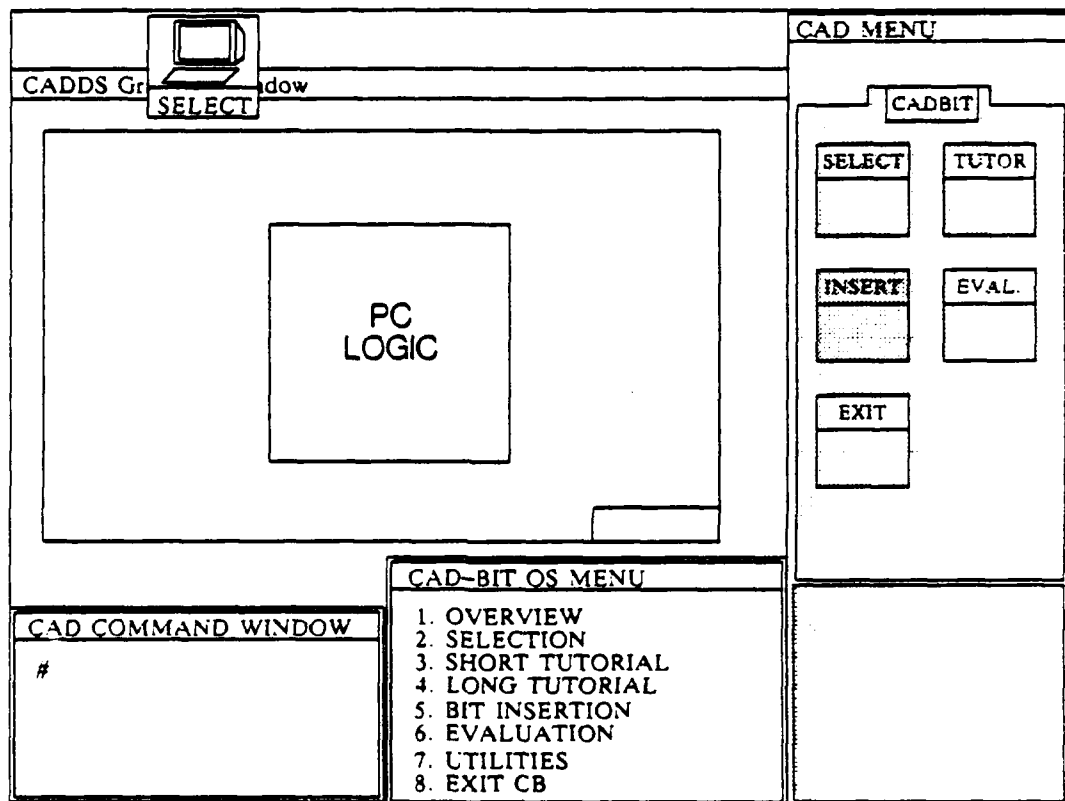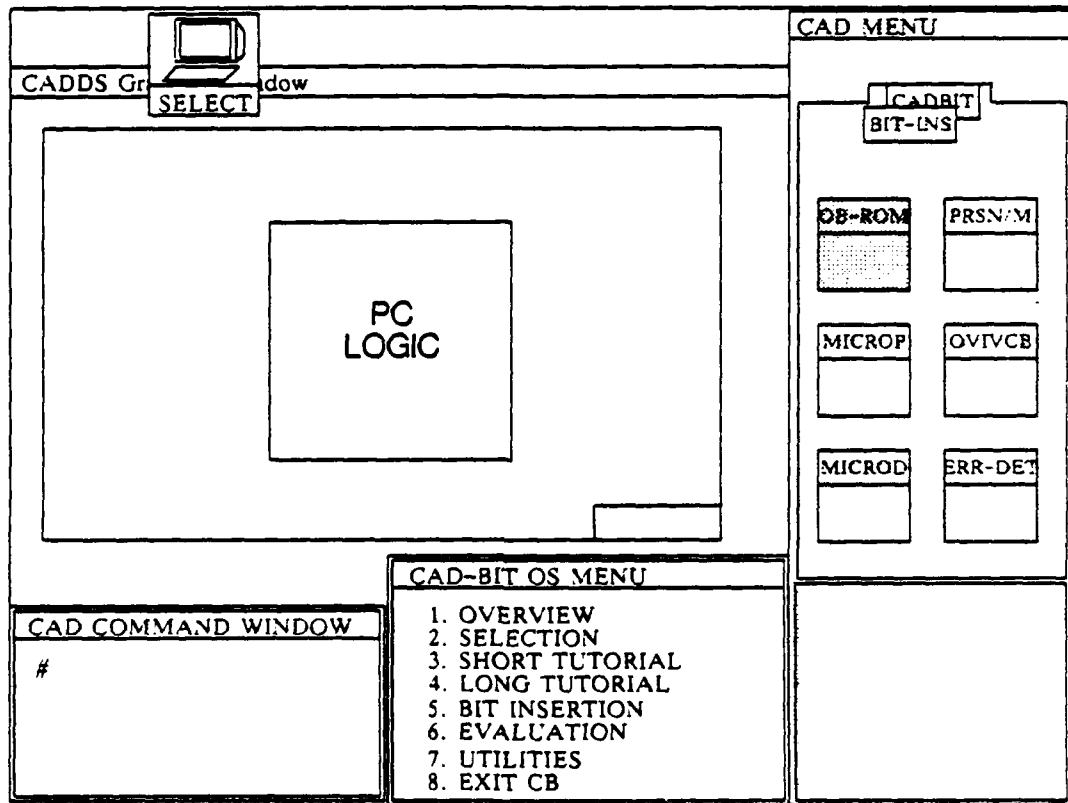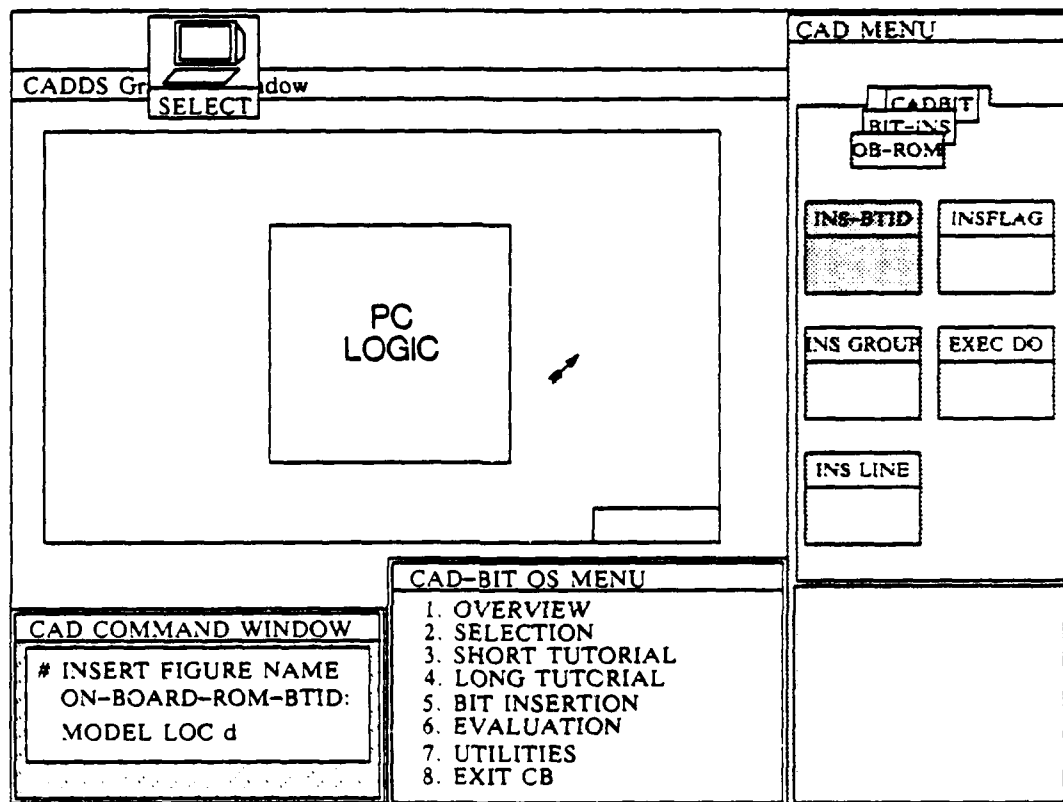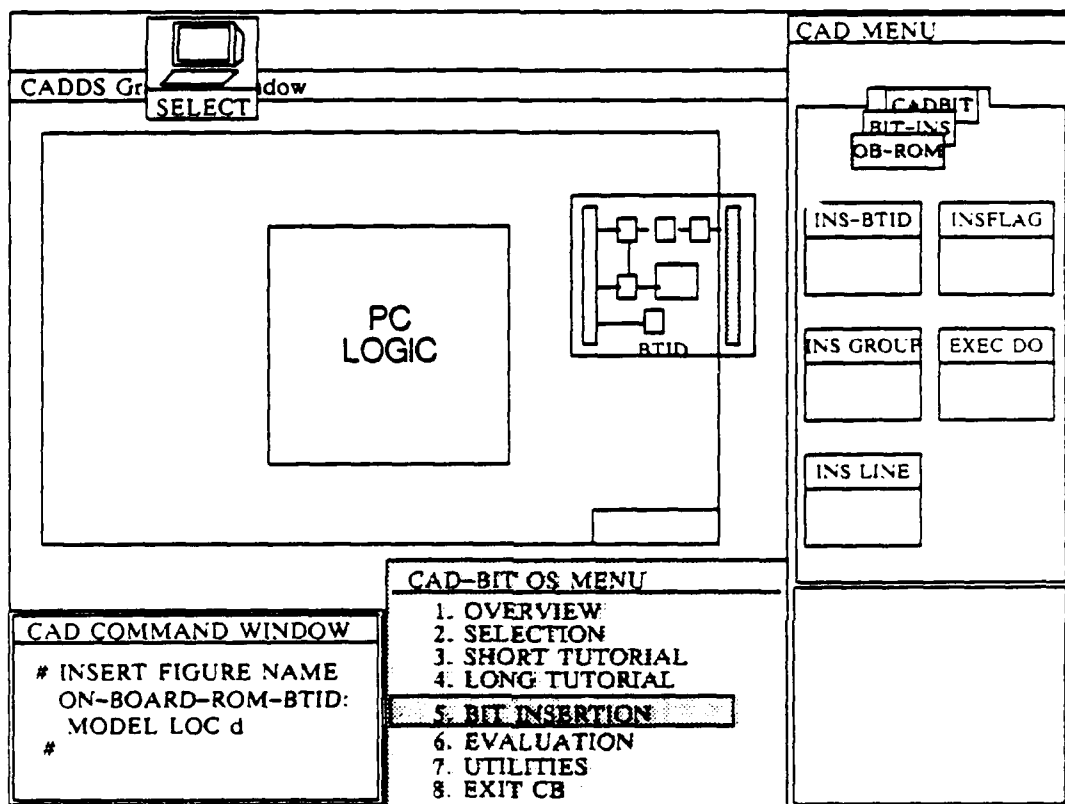
Figure 2-36

**BTID INSERTED IN PCB DRAWING SPACE
IN CAD ENVIRONMENT:
OS MENU BIT INSERTION SELECTED**

2.6.2 COMPONENT INSERTION PROGRAMS

After the BTID has been inserted, activities move to the CAD-BIT operating system environment. As each BTID BIT circuit is selected for insertion, a program is run which computes the quantity of each circuit type. The circuit quantity, the CAD host system syntax file, and host system component library names are used to generate the CAD command for circuit insertion. The resulting CAD command is referred to as the "DO" command. Figure 2-37 illustrates this process. The generated command is placed in a fixed name execute file. The user generates that fixed command to execute the insertion command with the proper CAD system syntax and component figure name.

Figures 2-36 and 2-38 through 2-40 provide a detailed step by step description of the command generation. In Figure 2-36 the BIT INSERTION menu item is selected from the CAD-BIT OS MENU. This results in the display of a BIT techniques menu in the same window, Figure 2-38. The selection of a BIT technique (On-Board ROM) from this menu results in that technique's circuit list (refer to paragraph 3.2.2) being displayed as in Figure 2-39. For each item in the menu list there is a corresponding circuit box in the BTID. The selection of a circuit (ROM in this example) from this menu list executes a program which generates the CAD circuit insertion command in the host CAD system syntax. This selection also generates a message in the CAD-BIT OS window to execute this command from the CAD environment menu and informs the user of the component and quantity which is to be inserted. This is shown in Figures 2-40 and 2-41.

2.6.3 "DO" COMMANDS

The generated command described above is referred to as the "DO" command. The user generates the "DO" command for the circuit type selected in the technique circuit list. Then the user executes the command by selecting the "EXEC DO" icon (Figure 2-40) from the CAD menu. The selection of this menu item results in the CAD command being issued, refer to the CAD COMMAND WINDOW of Figure 2-42. The arrows indicate where the user wants the ROMs inserted. Figure 2-43 shows these circuit elements inserted into the PCB design. The user will "DO" it again for each circuit type in the circuit list and BTID by repeating the process begun in Figure 2-36.

2.6.4 CIRCUIT INTERCONNECTION

Circuit interconnections are made after the BIT circuit elements are inserted into the functional design using the host CAD system's standard commands. The BIT tutorials contain interconnection application information to aid this process.

T01 : ON-BOARD ROM
T02 : PRPG/MISR
T03 : MICROPROCESSOR BIT

BIT TECHNIQUE LIST

CIRCUIT	DEFAULT P/N
1. COMPARATOR	LM319N
2. OP-AMP	MC1558
3. JK-FLIP-FLOP	74H103
⋮	⋮
#. ROM	TBP38516

TECHNIQUE CIRCUIT MENU

CIRCUIT	QUANTITY
ROM	4
MUX	1
⋮	⋮
COMPARATOR	1

CIRCUIT QUANTITY FILE

CIRCUIT	DEF. P/N	LIBR. NAME
COMPARATOR	LM319N	CADBIT.LM319n
OP-AMP	MC1558	CADBIT.MC1558
JK-FLIPFLOP	74H103	CADBIT.74H103
⋮	⋮	⋮
ROM	TBP38516	CADBIT.TBP38516

DEFAULT PARTS FILE

SCREEN OUTPUT

USING 4 1024X8 ROMS P/N TBP38516
PLEASE DIGITIZE 4 LOCATIONS

RESULT

GENERATES COMMAND IN HOST CAD SYNTAX TO
INPUT 4 1024x8 ROMS OF PART NUMBER TBP38516

In CV SYNTAX

INSERT NFIGURE CADBIT.TBP38516
             MODEL LOC   dddd

SYNTAX WILL BE DIFFERENT FOR OTHER
CAD SYSTEMS.

"DO" COMMANDS

CAD SYSTEM	SYNTAX	TEXT
CV	INSERT NFIG	
MENTOR GR.	Activate Component	
DAISY	Component	
CV-SCHEM.	Get Symbol	

CAD HOST SYNTAX FILE

Figure 2-37
"DO" COMMAND GENERATION
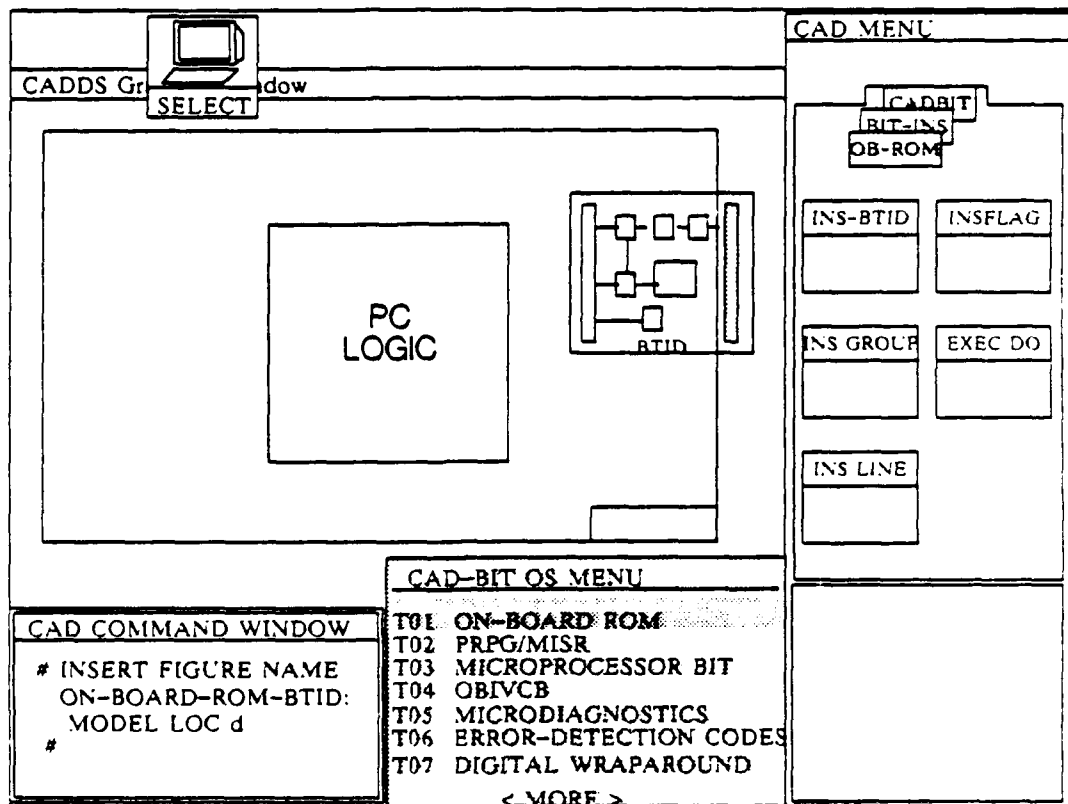
Figure 2-38

OS MENU BIT INSERTION:
TECHNIQUE SELECTED
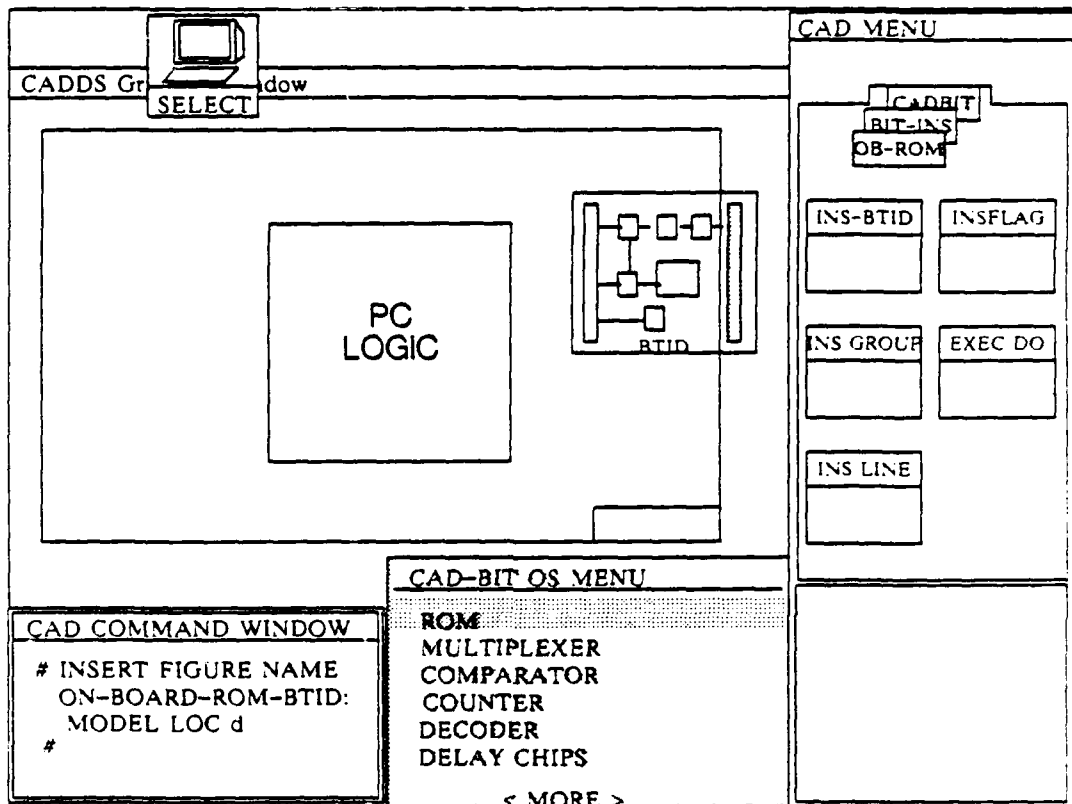
Figure 2-39

OS MENU BIT INSERTION:
CIRCUIT SELECTION
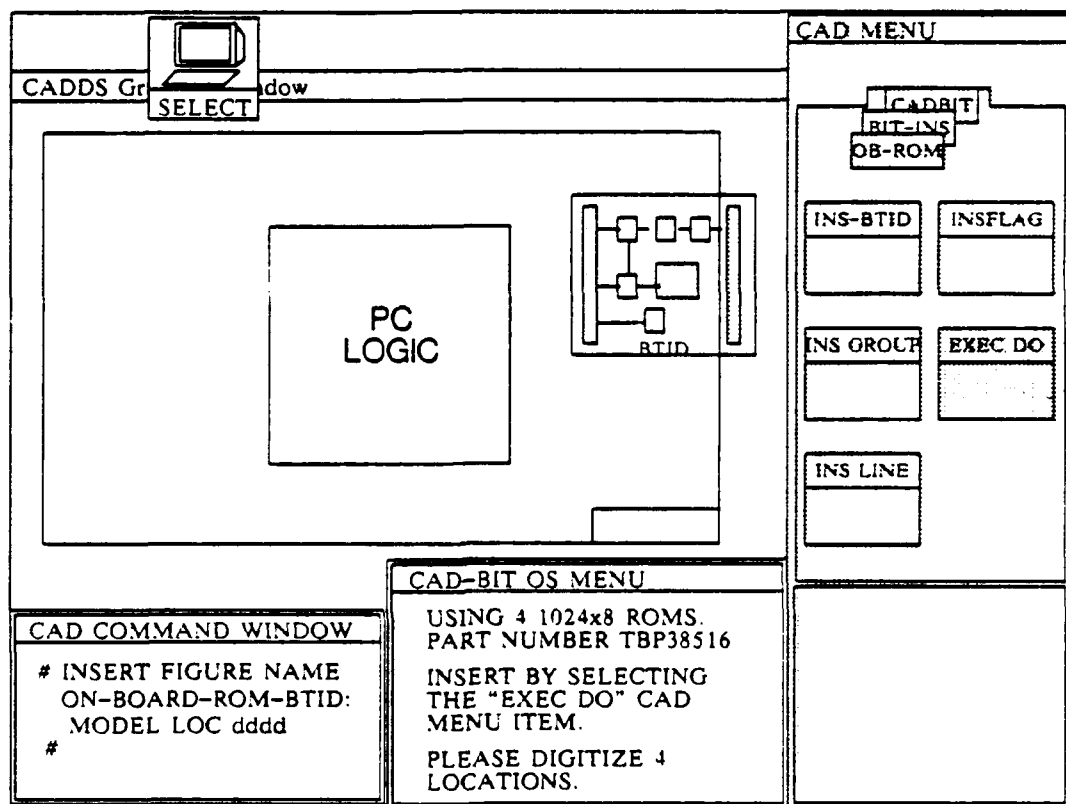
Figure 2-40

CAD ENVIRONMENT COMMAND EXECUTION
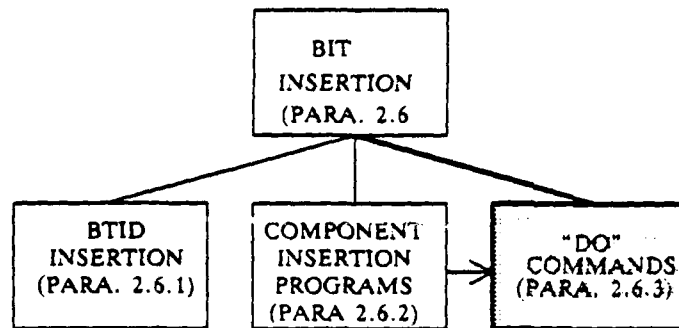
SCREEN OUTPUT

    Using 4 1024X8 ROMS P/N TBP38516

    PLEASE DIGITIZE 4 LOCATIONS

RESULT

    GENERATES COMMAND IN HOST CAD SYNTAX TO INPUT

    4 1024X8 ROMS OF DEFAULT PART NUMBER TBP38516.

    IN COMPUTERVISION CADDS 4X SYNTAX:

    INSERT NFIGURE SCH-LIB.TBP38516: MODEL LOC dddd

Figure 2-41
"DO" COMMANDS
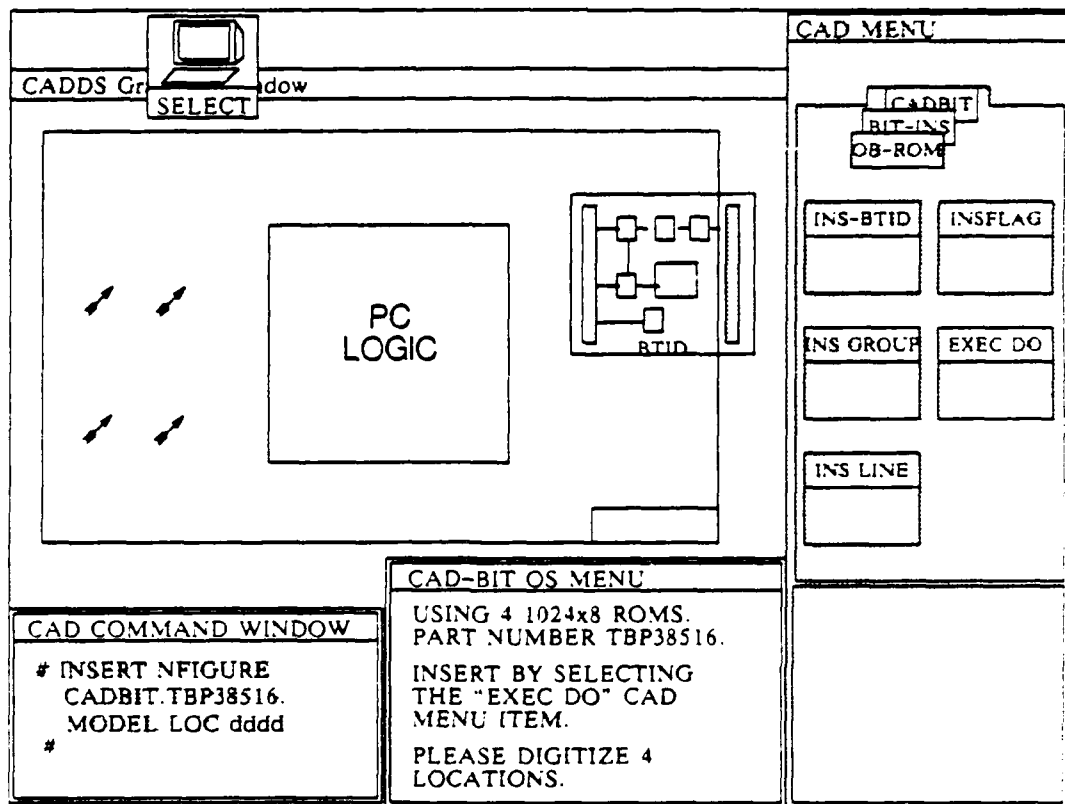
Figure 2-42

CAD MENU "DO" COMMAND EXECUTION
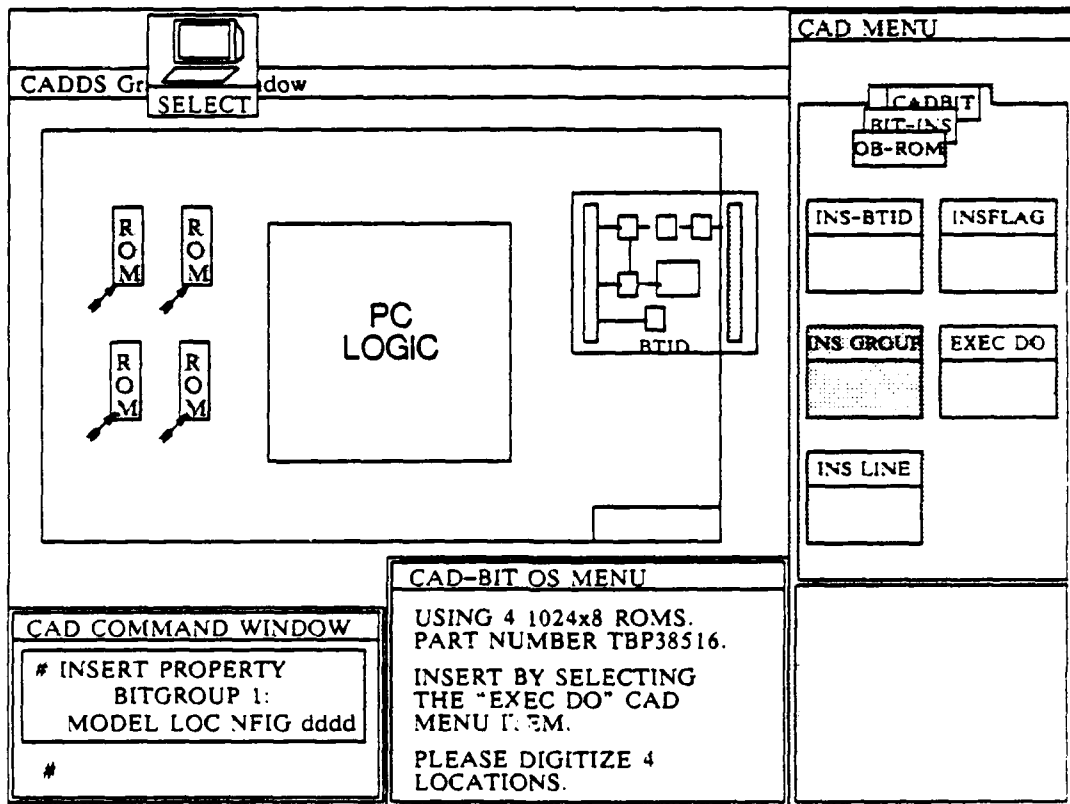
Figure 2-43

ROMS INSERTED: SELECT CAD MENU

INSERT BIT GROUP ATTRIBUTE ICON

2.6.5 BIT FLAG INSERTION

As BIT circuit components are added to the PCB design, the BIT circuits must be given two attributes which are required for the evaluation. The first is a BIT Group attribute so that particular components can be recognized as belonging to a particular BIT group. The second attribute assigns a decimal value 0 to 1.0 which defines the fraction of the component which is BIT related. If a component is totally used for BIT its BIT Factor attribute will be 1.0. If a component's usage is equally shared between the functional design and BIT, then its BIT factor will be 0.5.

Figure 2-43 shows the CAD menu icon "INS GROUP" highlighted. The selection of this item executes the CAD command shown in the CAD COMMAND WINDOW. The command inserts the BIT Group attribute (or property) and value (1) into the BIT components previously inserted. The arrows show the selected components to receive the BIT Group attribute.

Figure 2-44 shows the CAD menu icon "INS FLAG" highlighted. The selection of this item executes the CAD command shown in the CAD COMMAND WINDOW. The command inserts the BIT Flag attribute and value (1.0) into the BIT components. The arrows identify the selected components to receive the BIT Flag attribute.

2.6.6 BTID CHECK-OFF

Figure 2-45 shows the CAD menu icon "INS LINE" highlighted. The selection of this item executes the CAD command shown in the CAD COMMAND WINDOW of Figure 2-45. The command inserts a line across the ROM circuit element of the BTID to indicate that the insertion of these circuit elements is complete.

2.7 BIT EVALUATION

The evaluation is performed after the insertion of the BIT circuitry. It provides a report of the actual BIT penalties (power, area, etc.) incurred due to the insertion of the BIT circuitry. If multiple partitions have been used, each circuit group and its penalties will be shown separately. A summary of the penalties is also provided.

2.7.1 CAD-BIT ESTIMATION PENALTIES VS. EVALUATION

CAD-BIT provides an estimation of BIT penalties for each suitable technique to provide a penalty ranking for each technique. This aids the selection process. After selection and BIT insertion, an evaluation capability is provided to report the actual BIT penalties based on the actual circuit elements added as part of the BIT circuitry.
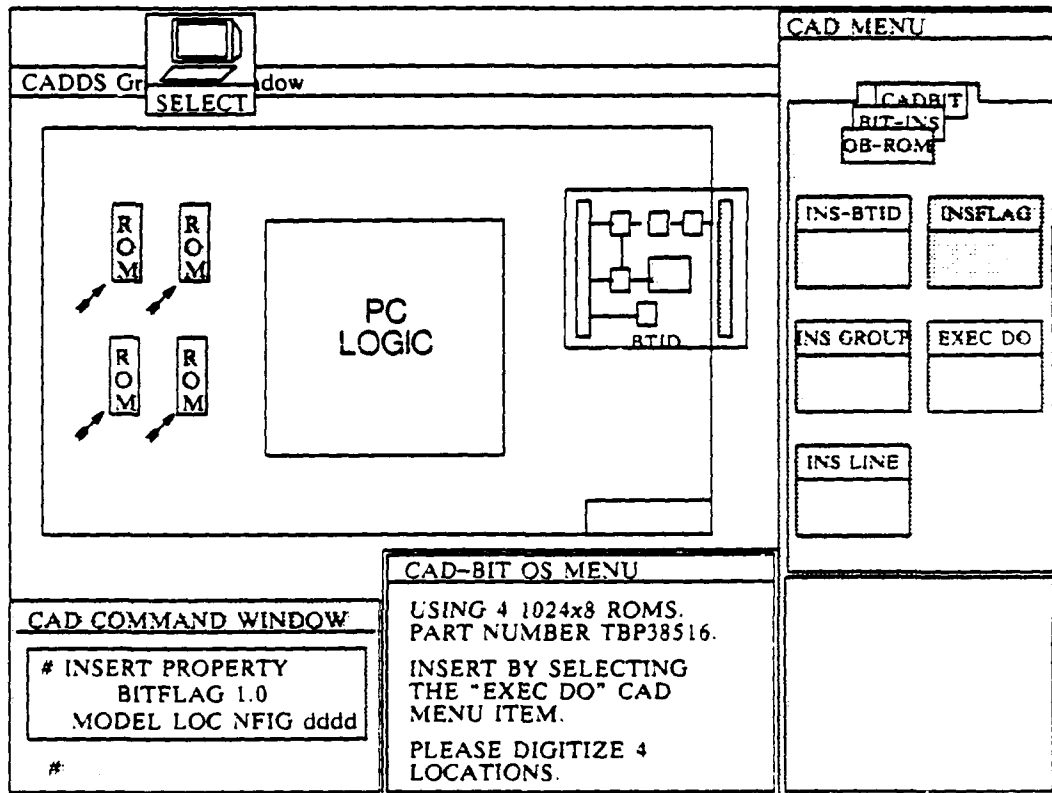
Figure 2-44
SELECT CAD MENU
INSERT BIT FLAG ATTRIBUTE ICON
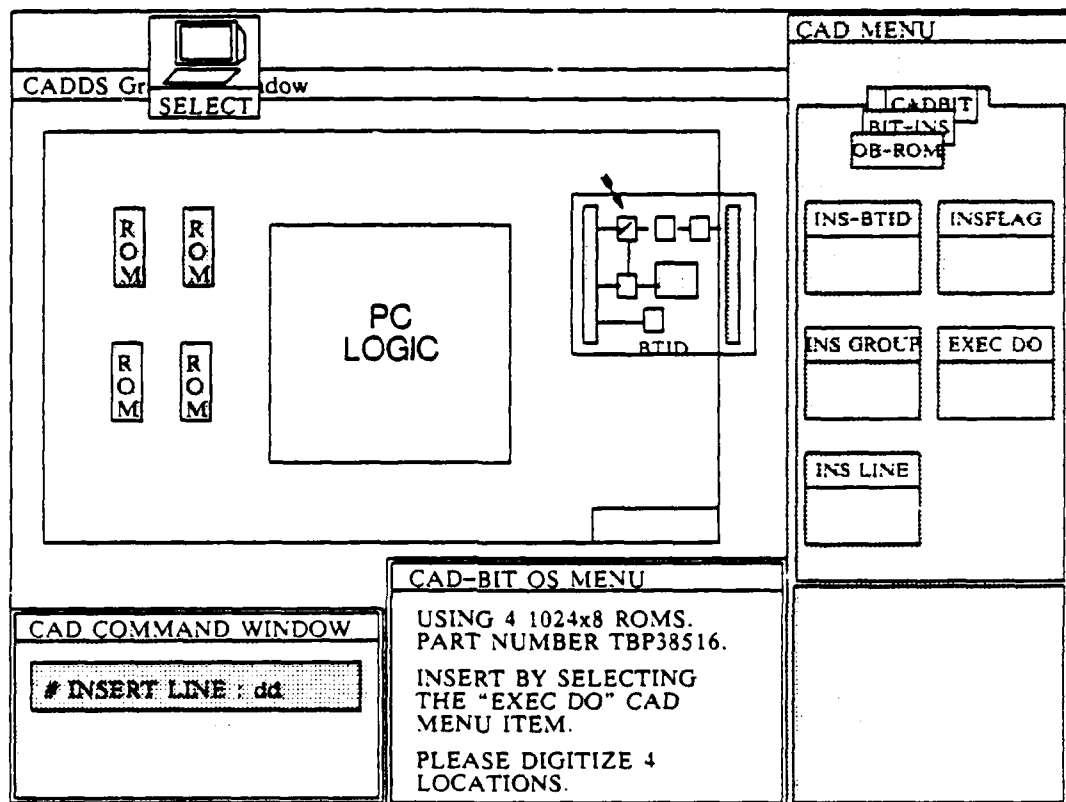
Figure 2-45

**BTID ROM CHECK-OFF:
LINE THROUGH ROM BOX IN BTID
(AT ARROW)**

2.7.2 BIT PENALTY EVALUATION

The BIT Penalty Evaluation is performed by extracting BIT circuit information under program control from the completed PCB design. The data extraction report is operated upon by the Evaluation Module using the Default Parts Penalty Data. The output from this module is the BIT Technique Penalty Report. This process is illustrated in Figures 2-46 and 2-47.

Since BIT Group data is inserted into the BIT components in the CAD PCB design, it is possible to distinguish the BIT penalties from different BIT techniques.

2.7.3 CAD SYSTEM DATA EXTRACTION PROGRAMS

The CAD system data extraction programs are CAD system dependent. They need to be generated for each CAD system by whatever tools are provided by the CAD vendor for this purpose. A universal program is not possible because each CAD system's data is maintained in a unique format.

2.8 UTILITIES

The Add-Technique, Fix-Penalty, and Generate-Circuit utilities are provided for BIT data base extensibility purposes. These utilities facilitate the adding of new BIT technique data. This data includes the associated insertion and penalty calculation programs, new suitability attributes and penalty parameters, and BIT circuit components and their associated penalty data. The effect which these utilities have on the CAD-BIT data base can be seen in Figure 2-48. The boxes enclosed with heavy dashed lines show those files which must be added when implementing a new BIT technique. This figure also shows how they are added.

2.8.1 ADD TECHNIQUE UTILITY

The Add-Technique Utility facilitates the addition of new BIT techniques. This utility creates a new sub-directory for the new technique, appends the new technique name and number to the Techniques List, and creates the technique attribute file used for suitability determination.

2.8.2 FIX-PENALTY UTILITY

The Fix-Penalty Utility generates a C program for the specified technique. The program contains formulas to compute the quantities of each circuit type, and the penalties for the BIT technique. The Utility also adds a call to this program into the main calling program. Fix-Penalty generates C language source code, compiles and loads the program.
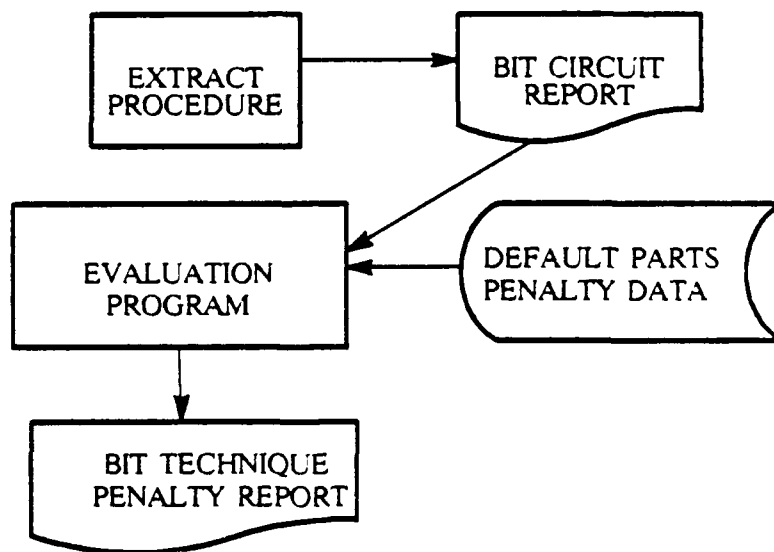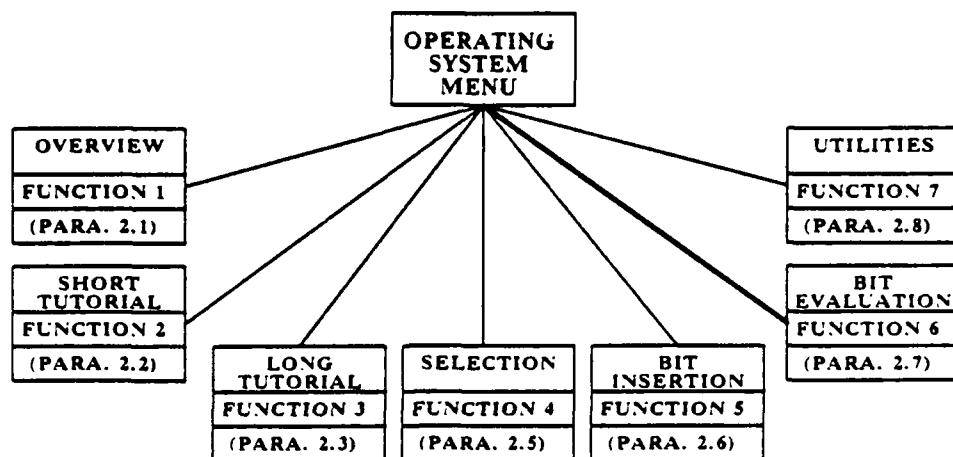
Figure 2-46

BIT EVALUATION

Figure 2-47

BIT PENALTY EVALUATION
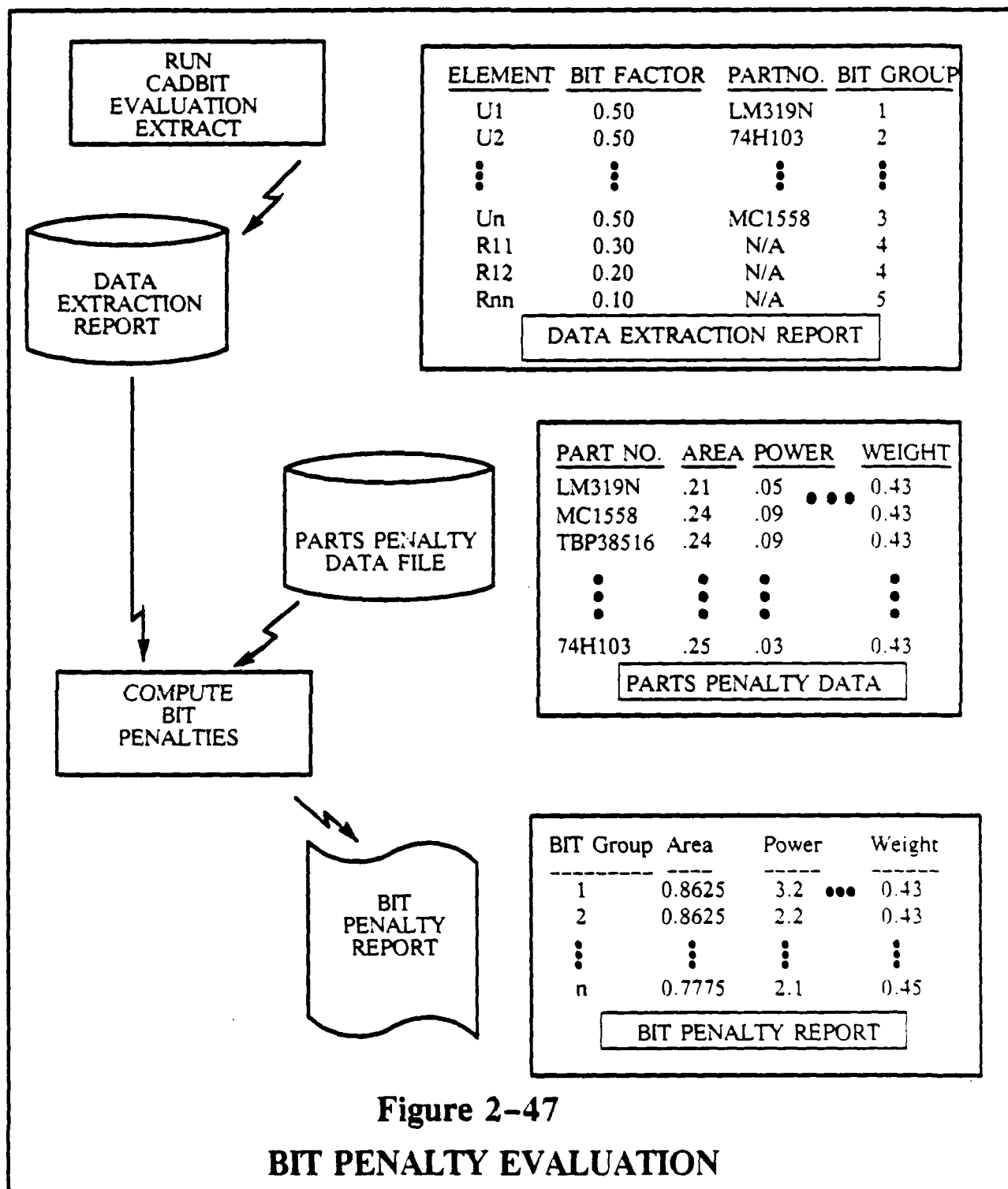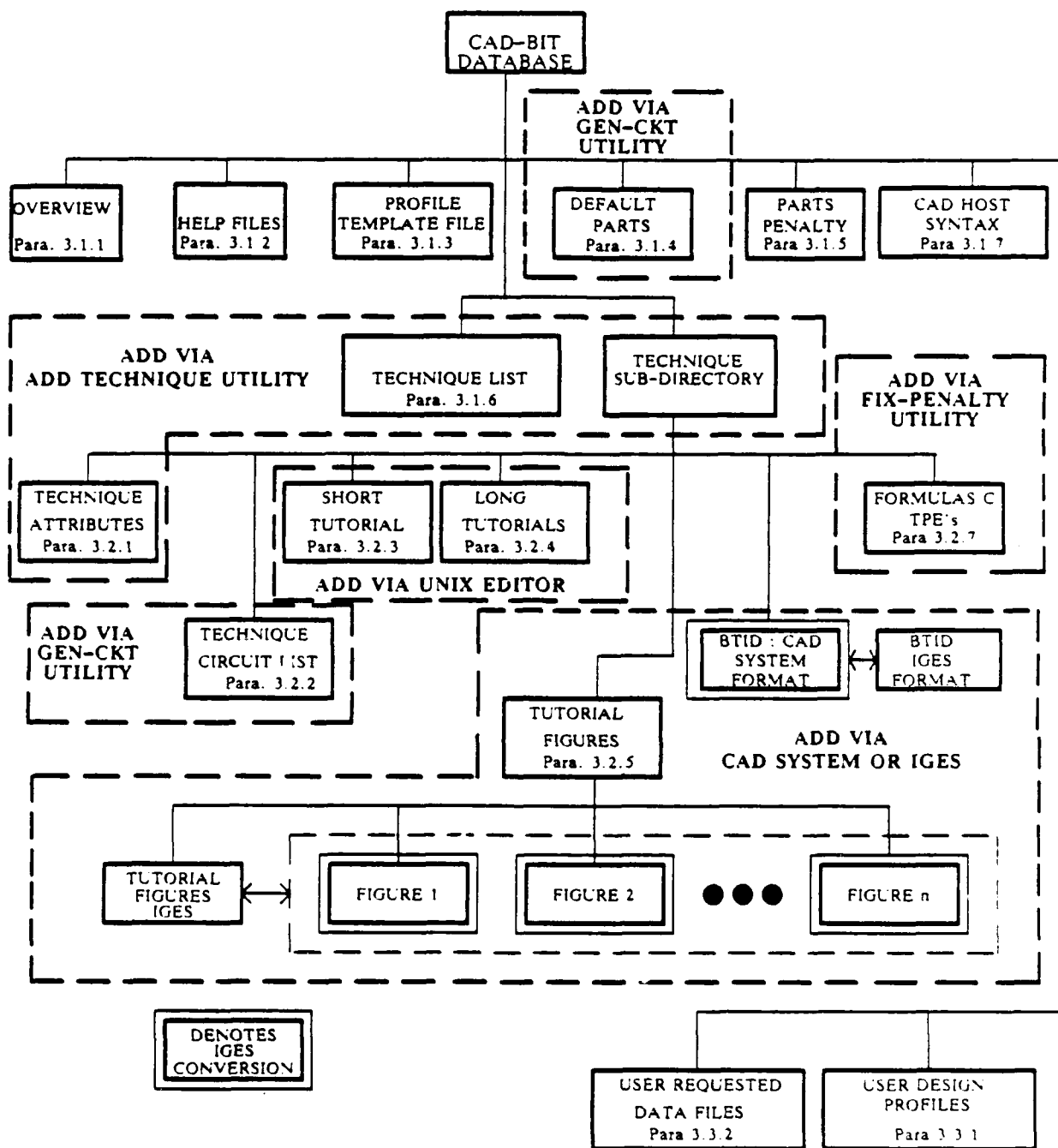
Figure 2-48

EXTENSIBILITY : ADDING TECHNIQUES

The C program is created interactively through the use of the utility and the worksheet shown in Figure 2-49. The worksheet allows the user to prepare for the interactive session. The data required to complete the worksheet is found in the Data Base Of Bit Functions. An example is included in Appendix A.

2.8.3 GENERATE-CIRCUIT UTILITY

The Generate-Circuit Utility generates the technique-circuit-menu file which is used by the BIT insertion module. This Utility also generates additions to the Default Parts File if new circuit elements are required. This program is interactive and ensures that the proper parts data will be present when a new BIT technique is added.

Sample Penalty Formula Generation WorkSheet
For On–Board ROM Technique

PART 1A INTRODUCTION •••••

1. Enter technique number (if applicable) :_____t01_____

If altering a TPE file :

2. How many User Requested Data (URD) questions will be asked ?_____5_____

{ *see Data Base of BIT Functions for questions pertaining to particular technique (URD)*}

3. How many Component Determination Equations (CDEs) will be used ?___8___

{ *see Data Base of BIT Functions for formulas pertaining to particular technique* }

PART 1B USER REQUESTED DATA ••••

(These questions DO NOT include the Technique Penalty Equations (TPEs). They are gener-
ated PART 3)

QUESTION	VARIABLE ASSIGNMENT
1. How many primary input pins are used by the PCB's Operational Circuitry ?	$v1$
2. How many primary output pins ?	$v2$
3. How many test patterns are to be stored in the ROMs ?	$v3$
4. What is the test pattern application rate ?	$v4$
5. What is the estimated initialization rate ?	$v5$
6.	$v6$
7.	$v7$
8.	$v8$
9.	$v9$
10.	$v10$

Figure 2–49

Sample Penalty Formula Generation WorkSheet
For On–Board ROM Technique

PART 2 COMPONENT DETERMINATION EQUATIONS •••••

(These formulas DO NOT include the Technique Penalty Equations (TPEs))

CIRCUIT TYPE	QUANTITY	FORMULA
ROM	n1	$(v1/8) (v3/2048) + (v2/8) (v3/2048)$
MULTIPLEXER	n2	$v1/8$
COMPARATOR	n3	$v2/8$
COUNTER	n4	$v3/4$
DECODER	n5	$1 + (n4/3)$
DELAY-CHIPS	n6	2
JK-FLIP-FLOP	n7	1
CONTROL-GATES	n8	2

PART 3 TECHNIQUE PENALTY EQUATIONS

(see Data Base of BIT Functions for formulas pertaining to penalty parameters (TPEs))

PENALTY
PARAMETER FORMULA

AREA $1.15 ((.375\ n1) + (.87\ n2) + (.375\ n3) + (.243\ n4) + (.375\ n5) + (.375\ n6) + (.243\ n7) + (.243\ n8))$

POWER $(325\ n1) + (350\ n2) + (375\ n3) + (455\ n4) + (375\ n5) + (200\ n6) + (90\ n7) + (110\ n8)$

WEIGHT $1.1 ((6.5\ n1) + (7.5\ n2) + (6.5\ n3) + (2.0\ n4) + (6.5\ n5) + (60\ n6) + (2\ n7) + (2\ n8))$

TESTTIME $(v3) (v4) + v5$

DELAY   N/A

Figure 2–49 (continued)

3.0 CAD-BIT DATA BASE

This section describes the CAD-BIT data base. The structure is illustrated in Figure 2-48. The data base consists of various types of files including ASCII files and graphics figures in the host CAD system format. The data base consists of three types of data, System Related Data, BIT Technique Related Data, and User Design Profiles. These are described below.

3.1 SYSTEM RELATED FILES

The following CAD-BIT files are neither BIT technique related nor associated with user design profiles.

3.1.1 OVERVIEW

The overview gives the user a brief description of the CAD-BIT module. Its format is that of a standard UNIX text file. It is shown in Figure 2-3.

3.1.2 HELP

The purpose of the CAD-BIT help facilities is to give the user on-line documentation about the BIT technique attributes available and the weighting factors applied to the technique penalty calculations. A sample Help file is shown in Figure 2-21.

The help facilities are all under the help sub-directory and the files themselves have the structure /variable-help, where variable represents all the attributes found in the user design profiles. Penalty parameters are also represented in the help facilities. They have the same file structure as the attribute help files.

The help facilities are standard UNIX text files containing a brief description of the attribute/parameter being accessed. Additional help files are added via the standard operating system editor by creating the new file and entering the explanatory information to be displayed.

3.1.3 PROFILE TEMPLATE FILE

The Profile Template File (PTF) is one of the key files in the determination of BIT technique suitability, BIT penalty estimation and technique ranking. The contents of this file are illustrated in Figure 2-19. The PTF is first used to create the User Design Profile. This profile is used to determine suitability by comparison of the User Design Profile attribute answers with the technique attribute files. In this process, the attribute list of the

PTF controls the comparison. For penalty calculations, the Penalty Parameter List is used as the master penalty list.

PROFILE TEMPLATE FILE CONTENTS

The Profile Template File consists of the three sections described below.

A. Suitability Attributes List

This section contains the list of attributes and associated data used to determine technique suitability. For each attribute, there is a corresponding suitability question, a set of allowable suitability question answers, and a default answer. The answers in the file are one letter abbreviations. For each abbreviated answer, the meaning of the one letter answer can be found in the associated attribute help file discussed above.

The Suitability Attributes are used to generate the User Design Profiles and to determine BIT technique suitability. The roles which the suitability attributes play in the generation of the User Design Profile and in the Suitability Determination are discussed in paragraphs 2.5.1 and 2.5.2, respectively.

B. Penalty Parameter List

This section contains the list of penalty parameters and associated data used to determine BIT penalties. For each penalty parameter, there is a corresponding penalty weighting factor question, and a default penalty weighting factor. The penalty weighting factors are numerical and each factor is relative to the others. A factor of 20 means that the penalty for that penalty parameter is twice the weight another penalty parameter with a value of 10.

The penalty parameters are used in computing penalties in the suitability determination phase. The roles which the penalty parameters play in the generation of the User Design Profile and in the Technique Selection are discussed in paragraphs 2.5.1 and 2.5.3, respectively.

C. Technique Override List

This section contains two modifiers to the technique suitability list. The two modifi-

ers are INCLUDE and EXCLUDE. Each modifier is used by the Profile Generator to create a list of techniques included or excluded from the final suitability list. These functions are discussed in paragraphs 3.5.1 and 3.5.2

PROFILE TEMPLATE FILE CHARACTERISTICS

The Profile Template File is a standard UNIX text file. Each category of data (SAL, PPL, and TOL) is identified within the file.

EXTENSIBILITY

Since the file is a standard UNIX text file, new attributes and penalty parameters are added via the standard text editor. To add an attribute or penalty parameter, one must also add the attribute or penalty parameter question. a default answer for the question, and for suitability attributes, a list of allowable answers (first characters only). A help file is also required to be added when implementing new suitability attributes or penalty parameters.

3.1.4 DEFAULT PARTS FILE

The Default Parts File contains information about the default parts which are inserted into the PCBs. This file is used by the BIT Circuit Insertion module (paragraph 2.6). The default parts file is a standard UNIX text file with fields containing the circuit name, part number, and the host CAD system library name, Figure 3-1. Additions to this file are made via the GEN-CIRCUIT utility. Since different CAD systems refer to circuit components by different file names, the Default Parts File will need to be modified for different CAD systems and will be part of the system initialization.

3.1.5 PARTS PENALTY DATA

The Part Penalty Data file contains part information such as part number, area. power. etc. This information is used during the evaluation of the circuits in the PCB. This file is a standard UNIX text file containing information on the penalties associated with each part in the file. The file is modified through the use of the standard editor. It is required that all parts used for BIT circuitry be present in this file for the evaluation process. If alternate parts are used in place of default parts. they must exist in this file. Figure 3-2 illustrates the contents of this file. See Volume II, Data Base of BIT Functions. for Part Penalty Data.

CIRCUIT	DEF. P/N	CAD FILE NAME
COMPARATOR	LM319N	CADBIT.LM319N
OP-AMP	MC1558	CADBIT.MC1558
JK-FLIPFLOP	74H103	CADBIT.74H103
●	●	●
●	●	●
●	●	●
ROM	TBP38L16	CADBIT.TBP38516

Figure 3–1
DEFAULT PARTS FILE

PART NO.	AREA	POWER		WEIGHT
LM319N	.21	.05	● ● ●	0.43
MC1558	.24	.09		0.43
TBP38516	.24	.09		0.43
●	●	●		●
●	●	●		●
74H103	.25	.03		0.43

Figure 3–2
PARTS PENALTY DATA

3.1.6 TECHNIQUE LIST

The Technique List file contains a list of all CAD-BIT operational BIT techniques. It is used for program control, user reference, and for menu display. This file is a standard UNIX text file containing two columns of information: the technique name and technique number. This is illustrated in Figure 2-23. Techniques are added to the list via the add-technique utility. Refer to the Utilities section 3.8.

3.1.7 CAD HOST SYNTAX FILE

The Cad Host Syntax File is illustrated in Figure 3-3. It contains CAD system names and corresponding CAD system syntax information. This file is used during the BIT Circuit Insertion phase to construct the CAD environment component insertion command in the host CAD system's syntax. This process is described in paragraph 2.6.

CAD SYSTEM	SYNTAX	TEXT
CV	INSERT NFIGURE	:
MENTOR GRAPHICS	ACTIVATE COMPONENT	
DAISY	COMPONENT	
CV CADDSTATION	GET SYMBOL	

Figure 3-3
CAD HOST SYNTAX FILE

3.2 BIT TECHNIQUE RELATED FILES

These are data files associated with the individual BIT techniques. The Technique related files refer to all files required for the use of the technique. This includes selection, insertion, evaluation, and the tutorial text and graphics figures used for those functions. This data can be found in Volume II, Data Base of BIT Functions, for each technique.

3.2.1 TECHNIQUE ATTRIBUTES

The Technique Attributes files are BIT technique files containing the attributes and the corresponding values required for them to be considered suitable for a particular PCB design. The BIT technique files are standard UNIX text files and are illustrated in Figure 2-24. Technique Attributes files are added when new techniques are added. They are added via the ADD-TECHNIQUE utility. That utility is described in paragraph 2.8.1

3.2.2 TECHNIQUE CIRCUIT LIST

The Technique Circuit List file contains a list of all the circuits and their default part numbers for a particular technique. These files are used as both menus and as inputs into the BIT Insertion module. The circuit lists are standard UNIX text files containing the circuit names and default part numbers pertaining to each BIT technique. A sample Technique Circuit List is illustrated in Figure 3-4. Circuit List files are added by running the Gen-Circuit Utility procedure. This utility's function is described in paragraph 8.5.1.

CIRCUIT	DEFAULT PART NUMBER
1. COMPARATOR	LM319N
2. OP-AMP	MC1558
3. JK-FLIP-FLOP	74H103
●	●
●	●
●	●
#. ROM	TBP38516

Figure 3-4
TECHNIQUE CIRCUIT LIST

3.2.3 SHORT TUTORIAL TEXT

The short tutorials are standard UNIX text files containing a brief description of the BIT techniques used in the CAD-BIT system. The short tutorials are added via the standard operating system editor. A sample short tutorial for the On-Board ROM technique is illustrated in Figure 3-5.

On-Board ROM Self Test is a non-concurrent Built In Test (BIT) technique which relies mostly on hardware and firmware. This technique is implemented by applying test patterns, which are stored in an On-Board ROM to a Circuit Under Test (CUT). The CUT's response is then applied to what is expected, resulting in a go/no-go output signal.

A big drawback of this technique is that the number of test patterns required to exhaustively test a function is proportional to the cube of the gates. However, this technique still has some potential since each test pattern can be individually and selectively determined. This results in maximizing the percentage of fault detection to the test pattern ratio.

Figure 3-5
SHORT TUTORIAL: ON-BOARD ROM TECHNIQUE

3.2.4 LONG TUTORIAL TEXT

The Long Tutorials are standard UNIX text files containing detailed descriptions of the BIT techniques used in the CAD-BIT system. The Long Tutorial consists of the following sections:

1. Bit Sequence Description

2. Advantages

3. Disadvantages

4. Attributes

5. Default Design

6. Parts Data Table

7. Bibliography

Examples of these files are found in Appendix A. The Long Tutorial Text may reference figures. These are found in the figure files included in the BIT Library, Volume II.

3.2.5 TUTORIAL FIGURES

The tutorial figures include any type of figures which aid in the selection, insertion, and interconnection processes. They can be brought up in the CAD host graphics environment and are meant to be used with the tutorial text. These are in the CAD system format. Any type of figure may be present (chart, flow diagram, circuit interconnection diagram, etc.). An example of a tutorial figure is found in Figure 3-6.

3.2.6 BIT TECHNIQUE INSERTION DIAGRAMS (BTID)

The BTID is a special tutorial figure. Inserted into the PCB prior to the insertion of the actual BIT circuitry, the BTIDs act as guides and check lists for the user to follow. They contain boxes representing the circuits involved which are arranged so as to guide the user on how to do the actual insertions. As each piece is inserted, its box in the BTID is checked off to mark the circuit as being completed. The BTID must be included for all techniques. An example is found in Figure 2-32.
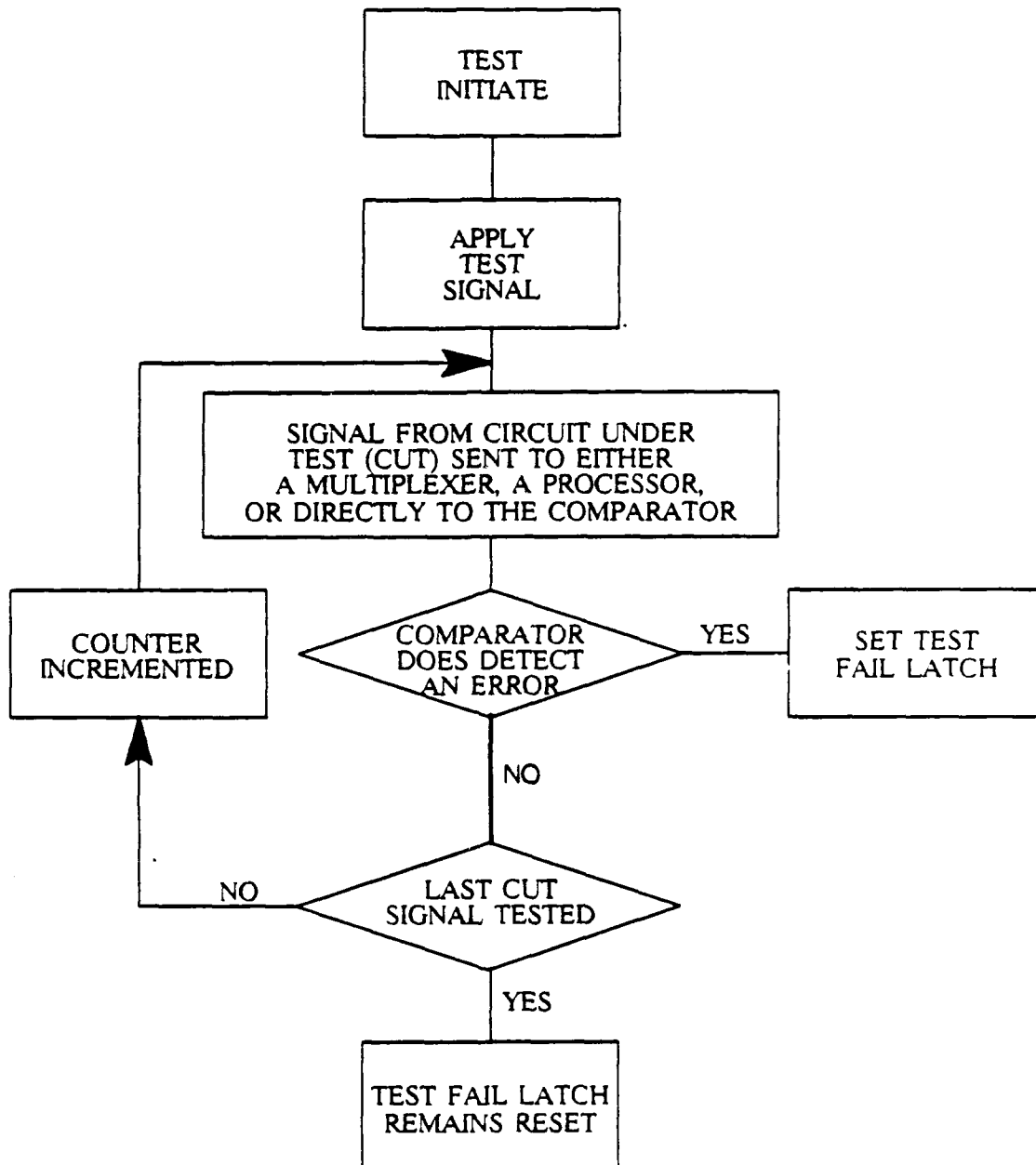
FIGURE 3-6
BIT SEQUENCE FLOW CHART EXAMPLE

3.2.7 TECHNIQUE PENALTY EQUATIONS

The Technique Penalty Equations (TPEs) are used during the Penalty Calculation portion of the Selection Module. The equations are incorporated into the C programs and are accessed when the penalties associated with that technique are calculated. These files are created via the Fix-Penalty utility, refer to Paragraph 2.8.2 and Figure 2-49. The source data for this may be found in Volume II, Data Base of BIT Functions.

3.3 DESIGN RELATED FILES

The Design Related Files contain data used by CAD-BIT pertaining to particular PCB designs. These files are not required after the design is complete and may then be deleted. These files consist of the User Design Files and User Requested Data Files.

3.3.1 USER DESIGN PROFILES

The User Design Profiles are files generated via the Profile Generation Module using the Profile Template File. These files are standard UNIX text files. This profile consists of the four sections below, as in Figure 2-20.

(1) IDENTIFICATION SECTION

The Identification Section is the first part of this file. The purpose of this section is to identify the particular User Design Profile and the user.

(2) SUITABILITY ATTRIBUTES LIST

The Suitability Attributes List contains the list of suitability attributes and the corresponding attribute values. These attribute values are answers to the attributes' suitability questions found in the Profile Template File. The attribute values are one letter abbreviations. For each abbreviated answer, the meaning of the one letter answer can be found in the associated attribute help file.

The data in the SAL is used to determine which BIT techniques are suitable to the PCB design. The attribute values are compared to the Technique Attribute List files. The role which the suitability attributes play in the Suitability Determination is discussed in paragraph 2.5.2 .

(3) PENALTY PARAMETER LIST

The Penalty Parameter List contains the list of penalty parameters and associated penalty weighting factors for the particular PCB design or portion of the design. The penalty

parameters are used in computing penalties in the suitability determination phase. The role which the penalty parameters play in the Technique Selection are discussed in paragraph 2.5.3.

(4)   TECHNIQUE OVERRIDE LIST

The Technique Override List contains the user selected technique overrides for a particular profile. This list is used to force the listed techniques to be included or excluded from consideration and can be found in the user design profile. This list is contained in a standard UNIX text file. It is the last section of the user design profile.

3.3.2 USER REQUESTED DATA FILES

The User Requested Data Files contain the responses and variable names from the Suitability module. This gives the user the option to use previously supplied answers when rerunning the Suitability module.

4.0 CAD-BIT MENUS

CAD-BiT MENUS, describes the upper level CAD-BIT menus, Figure 2-1. There are two distinct menus CAD and Operating System environments. The Operating System menu items control the non-CAD functions. These, however, do have a bearing on the CAD menu functions by providing instructions (tutorials) and references to the CAD menus. They also set up the BIT circuit insertion process and complete the BIT evaluation process. The CAD-BIT CAD environment menu items are in addition to the standard CAD menus found on the CAD system. They often refer the user to the CAD-BIT operating system menu. Both CAD and Operating System menu descriptions and sequences of menu operations are found in section 2.

4.1 MENU TRANSPORTABILITY

4.1.1 OPERATING SYSTEM MENUS

The Operating System Menus are transportable across UNIX systems since they use standard UNIX features. (C-shells, and the C language).

4.1.2 CAD MENUS

The CAD menus are not transportable across different CAD systems. CAD system's menus and the menu generation utilities are part of the CAD application software, thus the menus are non-transportable. It will be necessary to create CAD environment menus separately because for each different CAD system functionality may vary.

4.2 MENU EXTENSIBILITY

CAD menus may be extended within the limitations of each CAD system. CAD systems may limit the number of menu items per menu or the number of menu levels.

4.2.1 ADDING TECHNIQUES

Techniques may be added in the same manner as the original menu generation.

4.2.2 ADDING SELECTION PARAMETERS

Selection parameters are contained in the CAD-BIT data files and are controlled by the CAD-BIT software. Parameters may be freely added without affecting the menus.

5.0 CAD-BIT FEASIBILITY DEMONSTRATION

A feasibility demonstration was created to develop and evaluate the concepts to be used. The demonstration served two purposes. First it provided a means of evolving from initial concepts of the operating scenario to a user-friendly scenario. In developing the operating scenario, a deeper understanding of how the CBM should be structured also evolved. Secondly it provides a means of evaluating what the CAD-BIT product will be like before it is produced.

5.1 HARDWARE AND SOFTWARE

The demonstration was created to run on a Computervision Corp. (CV) CADDStation system. This consists of Computervision's CADDS 4X software operating on their SUN Microsystems platform. The hardware used was a CV CADDServer and diskless node workstations as shown in Figure 5-1.

TO:
- CV CDS 4101 CADDS SYSTEM
- DEC VAX's
- DAISY CAE SYSTEMS
- SUN-BASED ATE SYSTEMS

MAIN ETHERNET

CV 31C CADDStation

COMPUTERVISION (CV)
CADDServer
SERIES 32S

SOFTWARE:
- UNIX
- CADDS-4X

CV 32S CADDStation

ETHERNET SUBNETWORK

Figure 5-1
DEMONSTRATION HARDWARE AND SOFTWARE
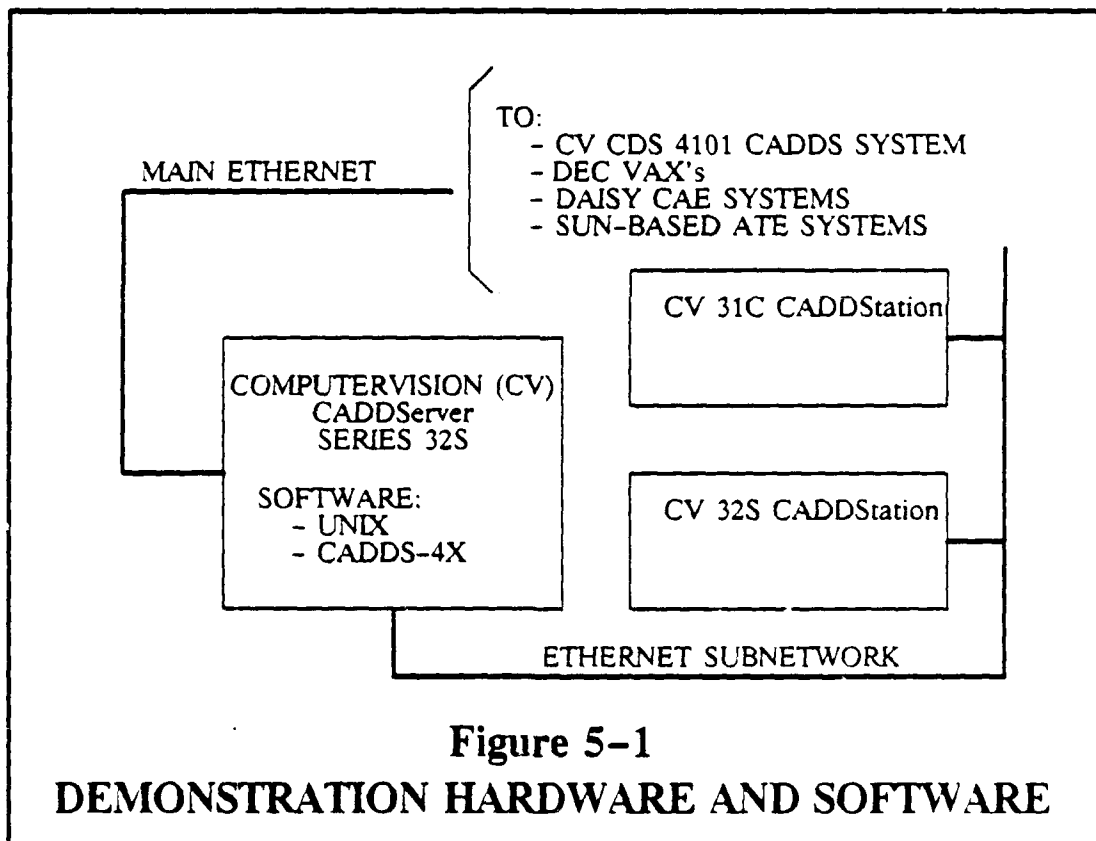
The demonstration was originally written with some software embedded into the CV CADDS application software. This demonstration software was later removed from

CADDS and made to operate in the UNIX operating system environment since this is the direction in which the automated procedure design evolved.

5.2 LEVEL OF IMPLEMENTATION

The following paragraphs describe the level of implementation.

Two BIT Techniques, the On-Board ROM and the Voltage Summing, were each partially implemented. The Operating System and CAD menus were created and operational to the extent described below.

At the Operating System level all functions were operational to some degree. The Overview was fully operational and the short and long tutorials were operational to the extent that each could be demonstrated for a tutorial text file and the insertion of tutorial figures in the CAD environment. BTID's for the two implemented techniques were available to the point where they became nested.

The profile generation process included new profile generation, profile editing, and help files. In the suitability process, the profile could be modified to produce different suitability lists. A feel for the operation of the forced inclusions and exclusions could be seen, but was not implemented. A "wired" selection process was available.

The BIT insertion process was demonstrated. The generation of the "DO" command at the Operating System level was implemented. By moving to the CAD environment window, the generated CAD command was issued with the resulting insertion of the BIT circuit components. Menu operations for the BIT Flag and BIT Group attribute insertion were implemented to prepare for the evaluation step.

The evaluation process was demonstrated through the data extract in which the component data (part number, reference designation, Bit flag , and BIT group) were extracted.

A version of the Fix-Penalty utility was demonstrated for the generation of the CAD component insertion command following the procedure described earlier in this report. The utility generated the C language source code, computed the required number of ROMs required, and generated the component insertion command using a CAD host syntax file.

All features demonstrated were via the CAD or Operating System menus. The interface between environments proceeded smoothly showing the benefits of the implementation on an operating system supporting windows.

5.3 CAD-BIT DEMONSTRATION SCREENS

Figures 5-2 through 5-7 illustrate the demonstration start up process using a sequential set of screens from login through the activation of both the CAD and Operating System menus.

LOGIN : cadbit

Figure 5-2
CAD-BIT LOGIN SCREEN
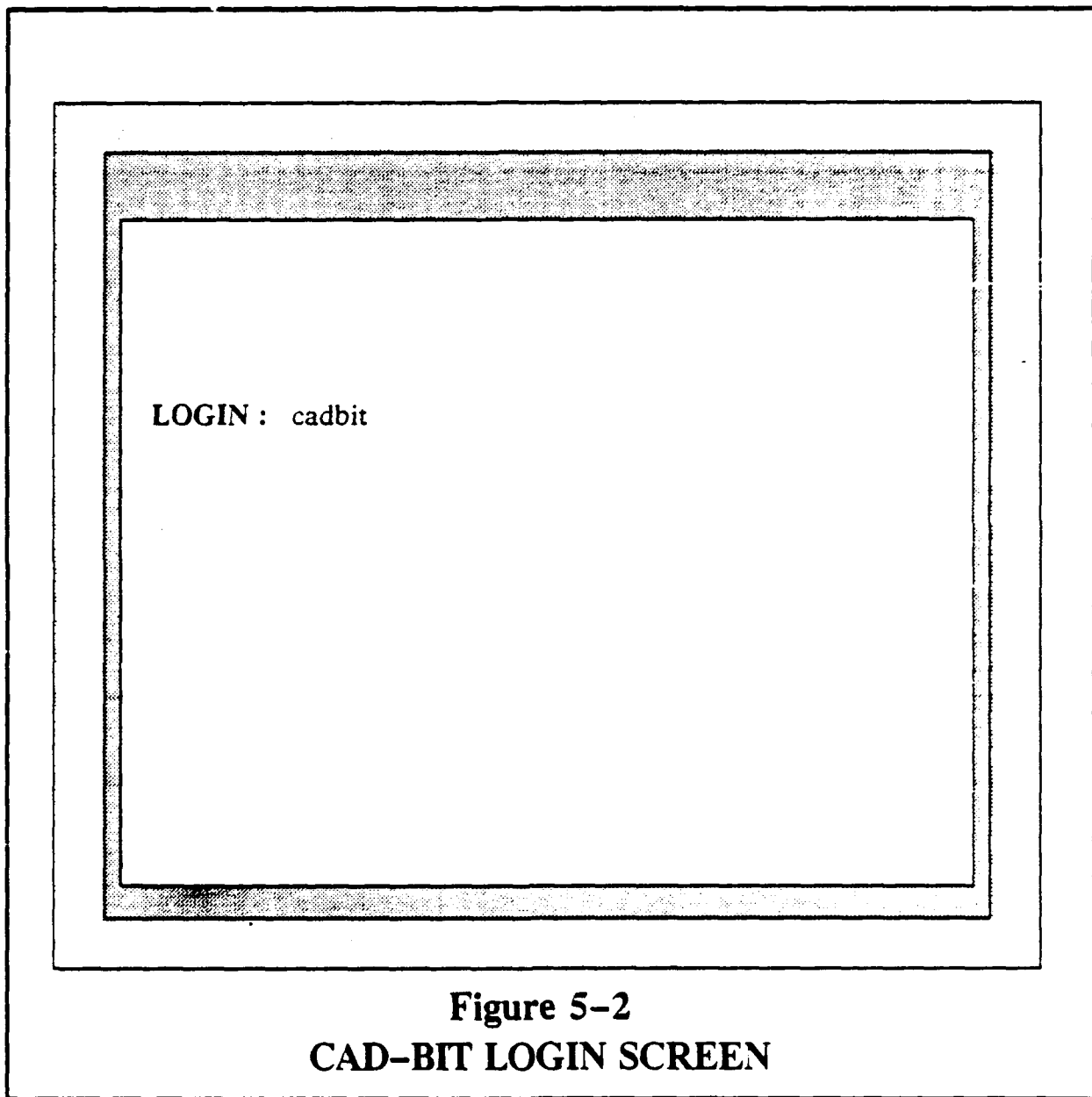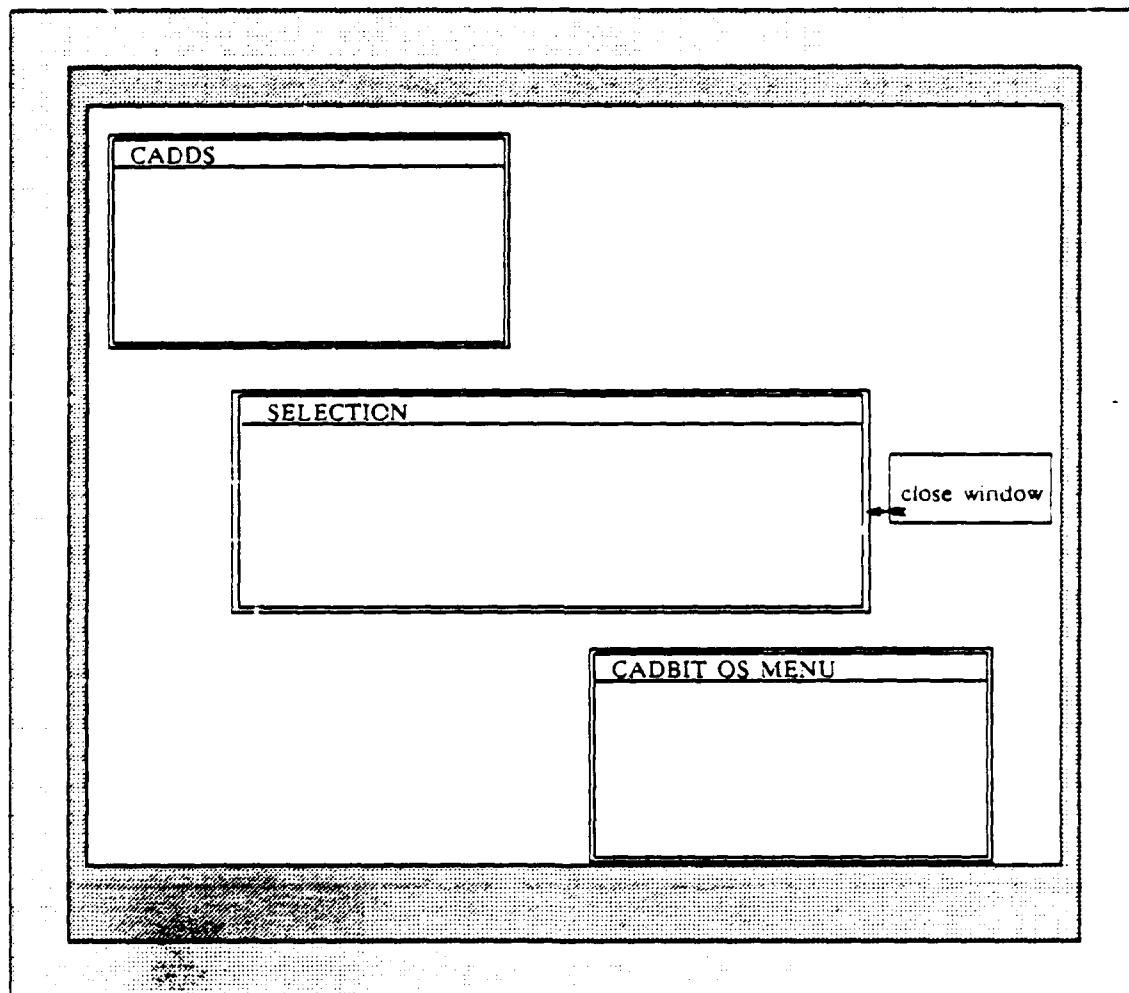
Figure 5-3
CAD-BIT MODULE SELECTION MODE SCREEN
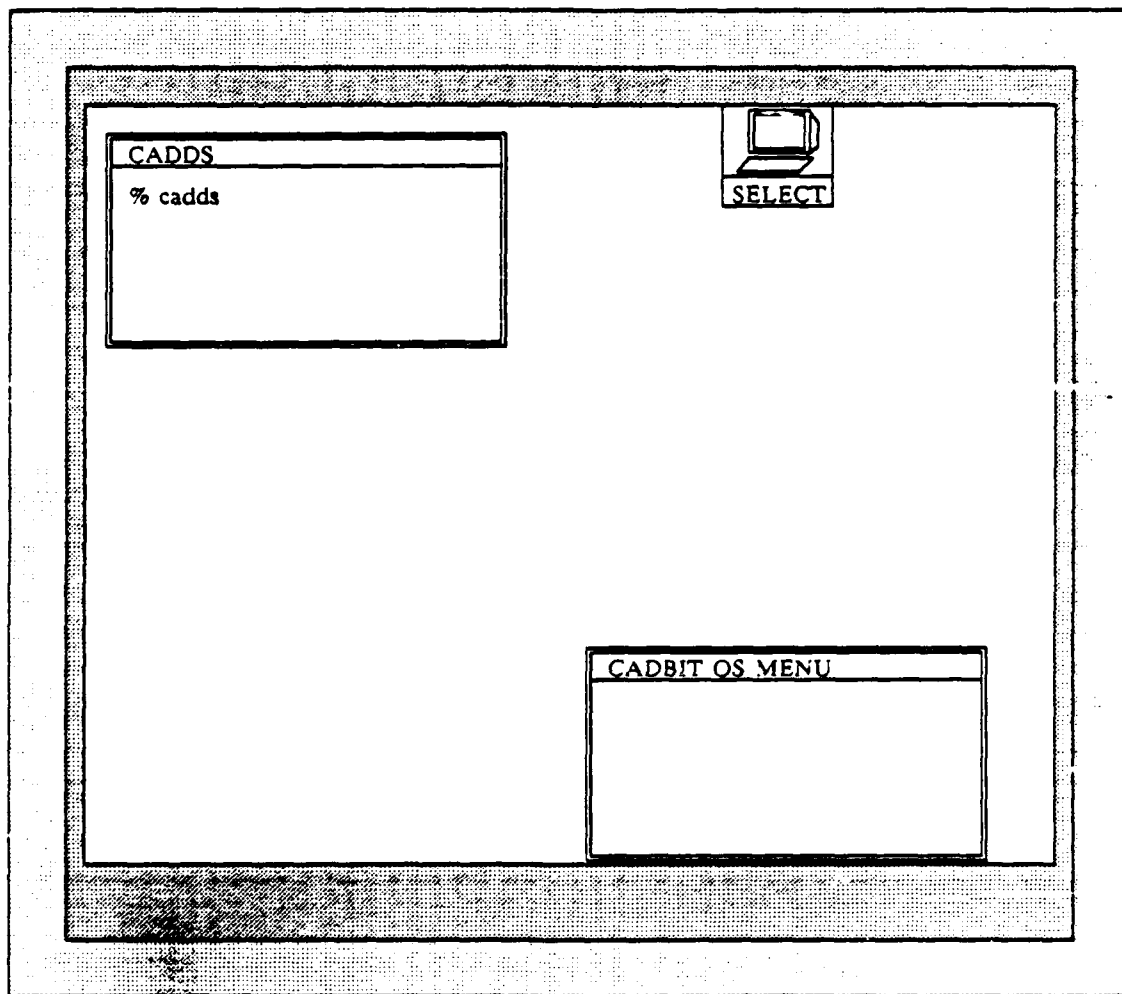
Figure 5–4

CAD–BIT MODULE GRAPHICS ENVIRONMENT SETUP
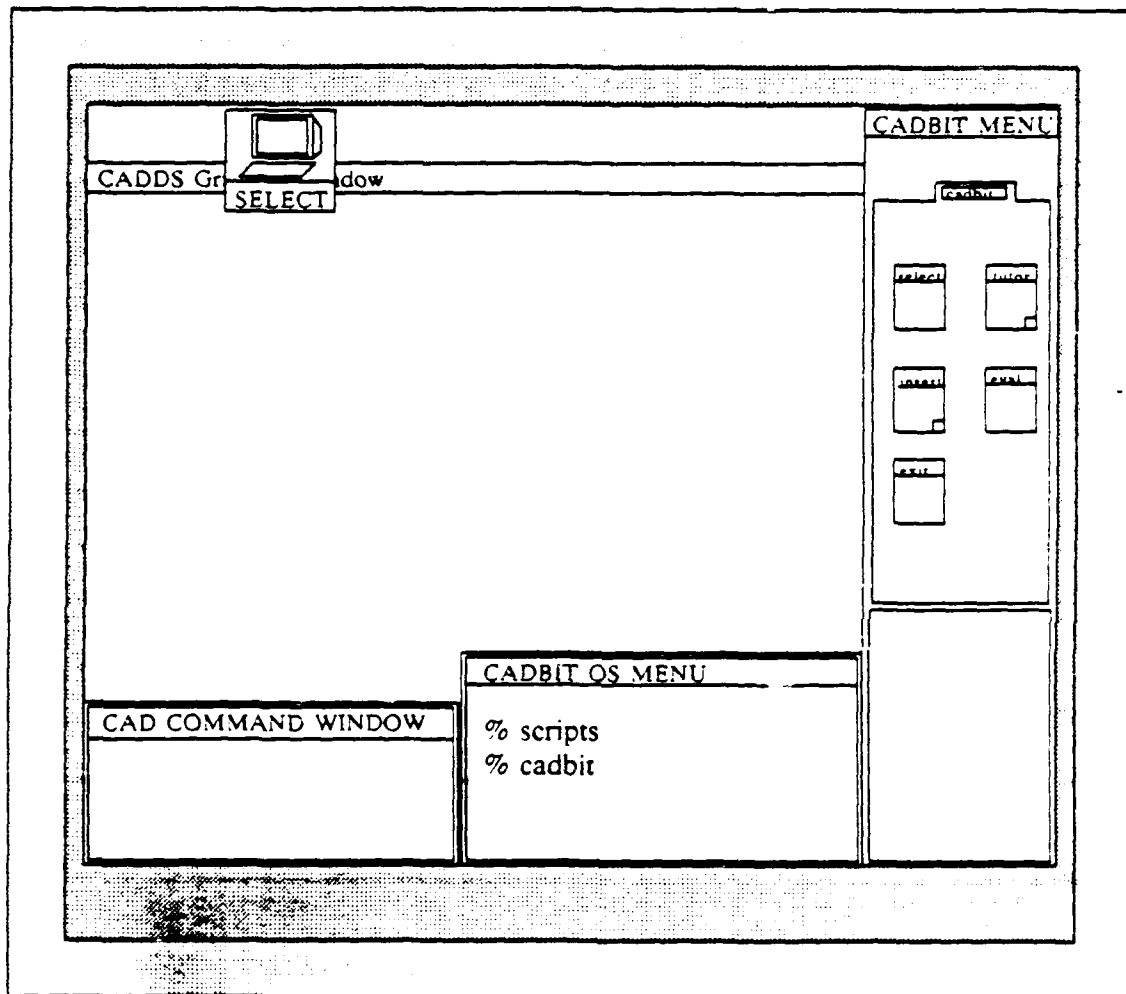
Figure 5-5

CAD-BIT MODULE OPERATING SYSTEM MENU SETUP
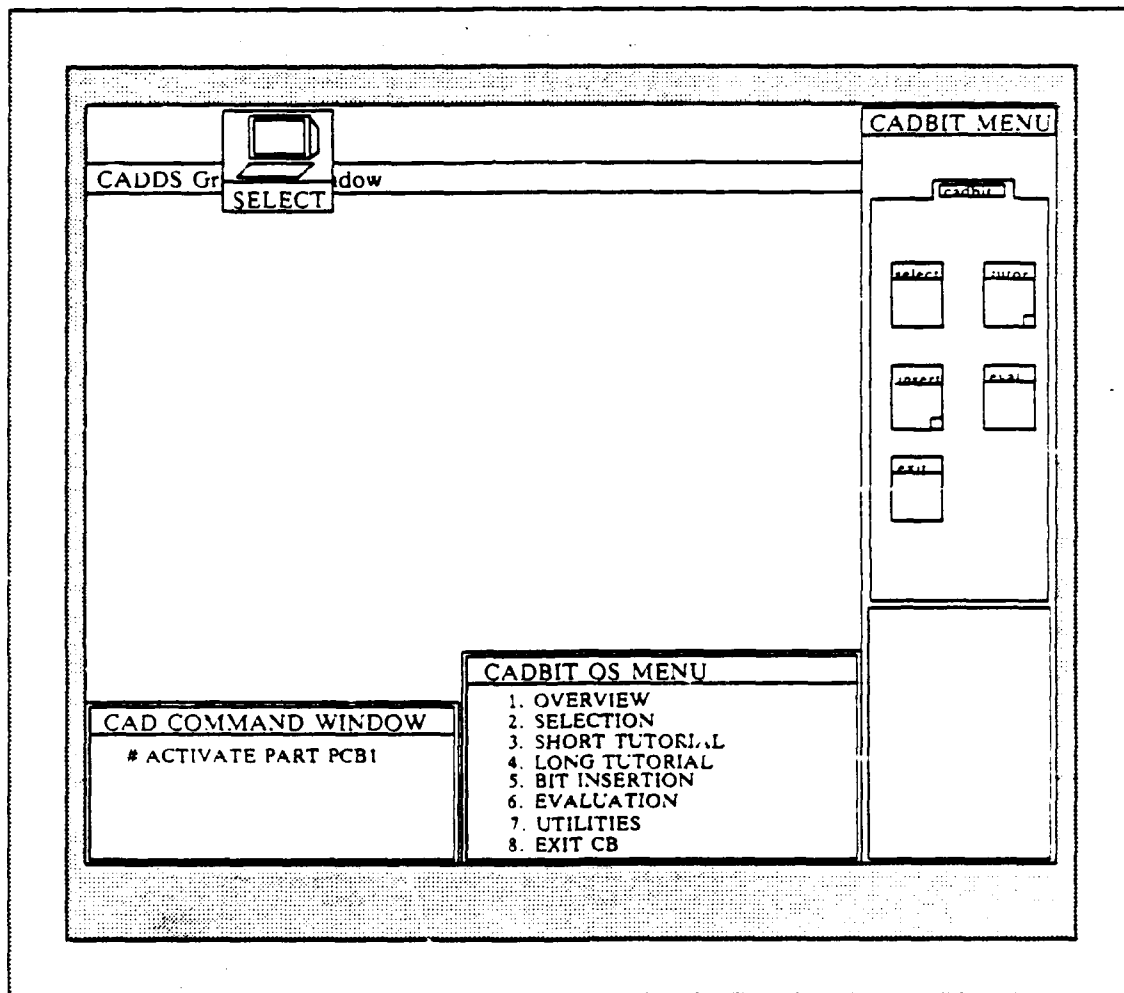
Figure 5-6

CAD-BIT MODULE OPERATING SYSTEM MAIN MENUS
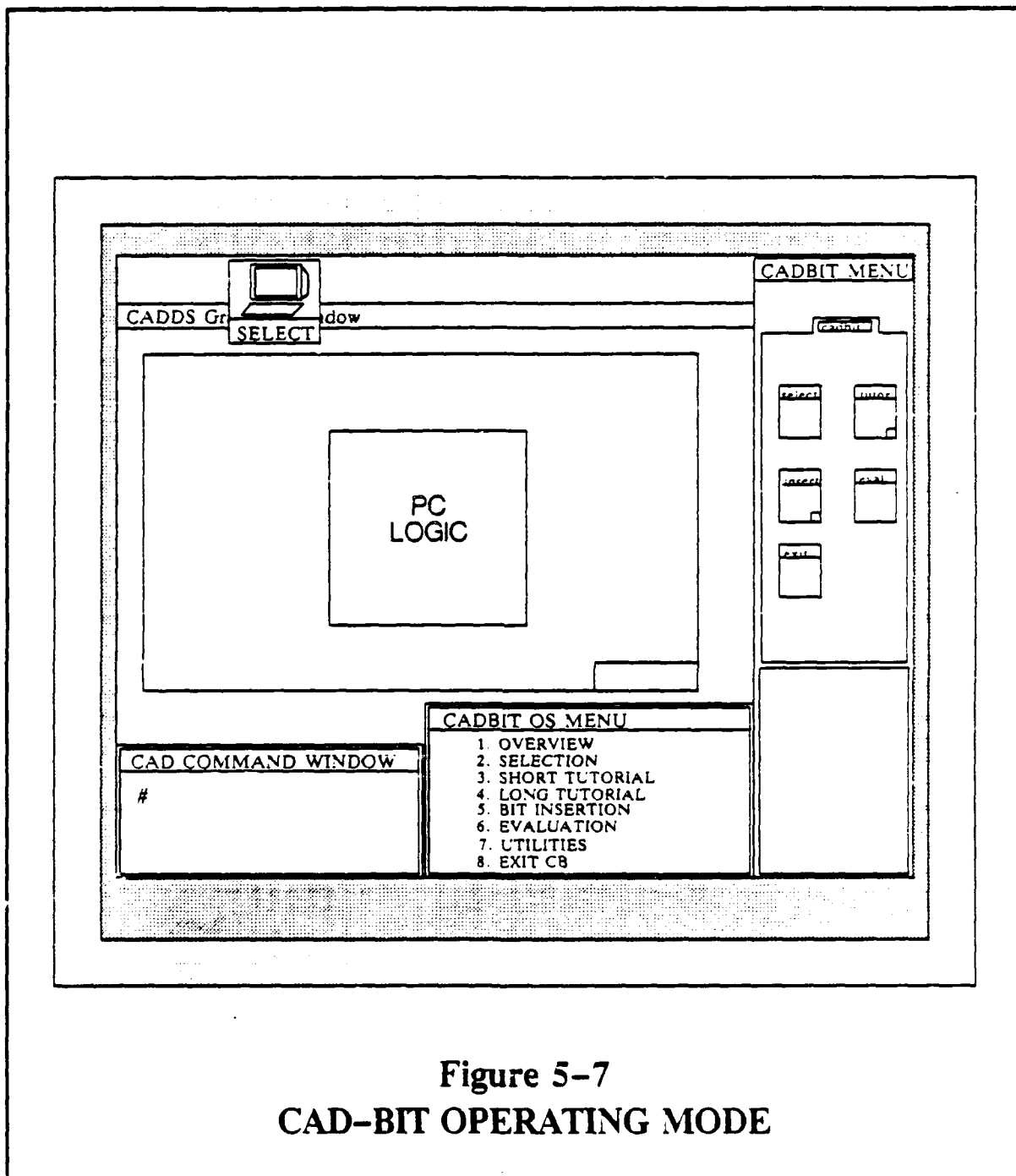
Figure 5-7

CAD-BIT OPERATING MODE

5.4 PORTABILITY

Portability of the feasibility demonstration is similar to what can be expected for the final CAD-BIT product. The operating system used was UNIX, and the language used was C. These can easily be ported to another UNIX/C system. The tutorial figures used could be converted to IGES using the system's resident IGES translator and ported to another CAD system which supports IGES. The data bases used by the C programs were ASCII files. Menus used in the CAD environment were CV CADDStation dependent.

6.0 BIT SURVEY

In order to obtain BIT techniques for insertion into the BIT Library, a BIT survey , performed. In the survey three different systems were examined - a digital, analog and hybrid system. The survey was augmented by reviewing several additional systems, performing a literature search and obtaining literature from two committees promoting testability buses. In addition, the BIT and Self-Test expertise residing within the company was tapped through conversations with knowledgeable personnel.

The following actions constituted the survey effort:

* Develop a list of BIT techniques for the BIT Library

* Identify attributes associated with each technique

* Analyze BIT structural alternatives

    - Complete stand-alone techniques

    - BIT Stimuli, BIT Responses, Embedded BIT

    - System Function vs. Board Functional BIT

* Perform tradeoffs among quantity of techniques vs. expandability of the data base.

6.1   SYSTEM SURVEY

The three systems, digital, analog and hybrid, are described below, respectively.

6.1.1   DIGITAL SYSTEM

The digital system selected for the survey is a computer from the JSTARS program. The system is Control Data Corporation micro Advanced Flexible Processor (uAFP+) It is a good example of a current system and employs a large amount of Application Specific Integrated Circuits (ASIC) in the form of gate arrays. Each gate array contains BIT circuitry which serves to check the gate array itself and perform coordinated tests with other ICs to verify connectivity. Figure 6-1 shows the BIT circuitry of a single gate array. The original Circuit Under Test (CUT) is shown shaded and the BIT circuitry is the unshaded blocks. With the use of this circuitry, the following tests can be performed:

* STATIC

* PSEUDO RANDOM SOURCE
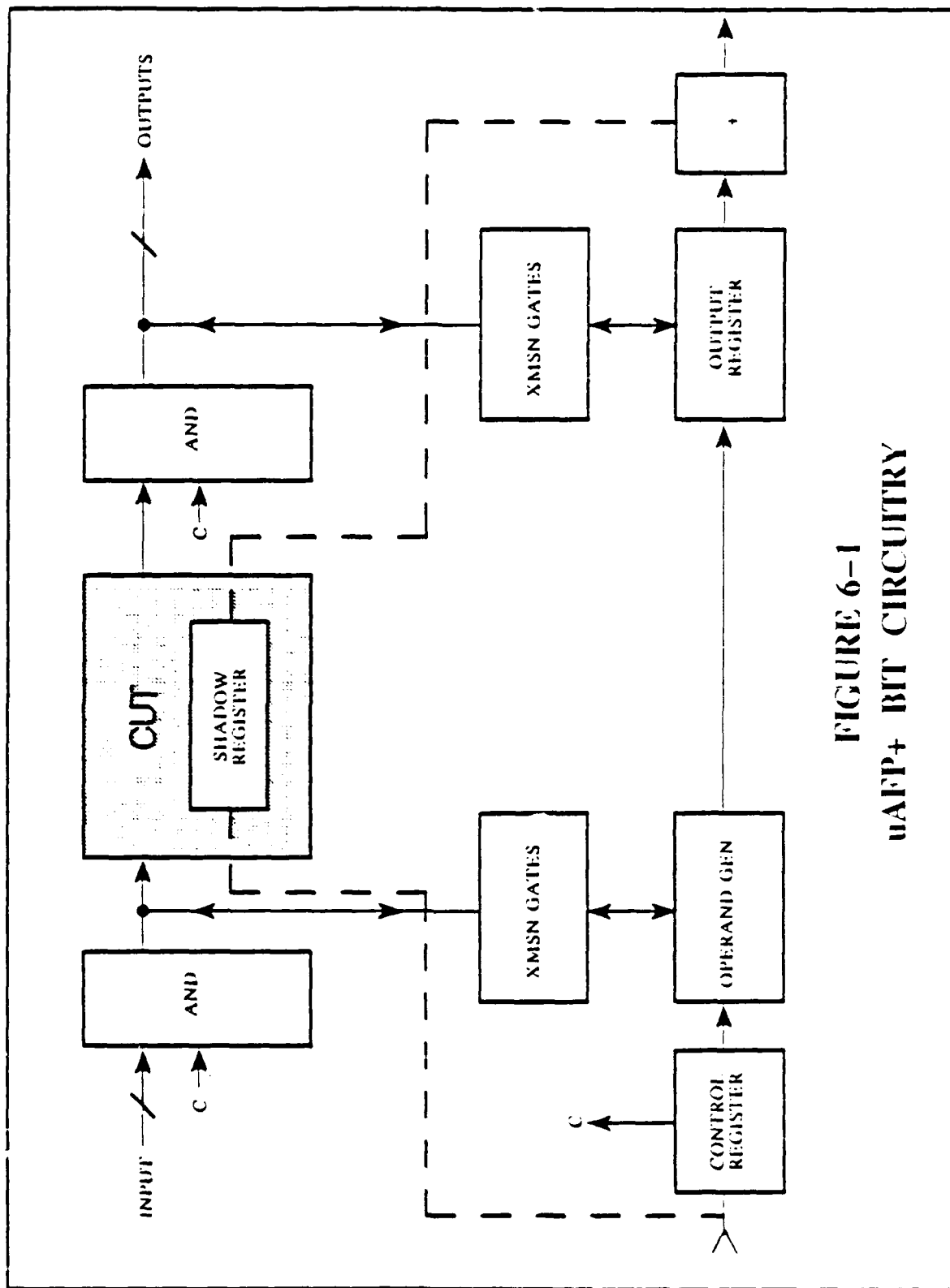
FIGURE 6-1
uAFP+ BIT CIRCUITRY

* CHECKSUM ON OUTPUT

* CONCURRENT DIGITAL

* INTERCONNECT

* SELF CHECK OF BIT CIRCUITRY

* SHADOW REGISTERS

Additionally, another digital system from the JSTARS program, the HAWK 32, was supplied by the Rolm corporation. This system was somewhat similar in its use of ASICs and BIT circuitry and served to reinforce the importance of this type of BIT in current systems.

6.1.2   ANALOG SYSTEM

The Head-Up Display of the F14/A6F was selected as a representative analog system. It is a current system, still under development by Kaiser Electronics. Figures 6-2 is an extract from the schematic of one LRA of the system. It is an example of the redundancy BIT technique. The circuit under test (CUT) is comprised of the XDEFL and YDEFL channels on the bottom of the diagram. The circuitry added for BIT is shown above the dashed line. This particular example shows that it is possible to avoid duplicating the entire CUT circuit in the resultant channels. For example, in the CUT channel. two inverting amplifier stages are cascaded to yield a non-inverted signal. The redundant channel uses just one inverting stage and sums this inverted signal with its non-inverted signal from the CUT to produce a net zero signal level which can then be monitored for any excessive deviation from zero. The actual circuit checks both X and Y deflection channels by feeding all of the signals into a four input summing amplifier. The error monitor is the window comparator circuit whose output drives the pass/fail line. This demonstrates the redundancy technique included in the BIT library. Also, the scheme utilizes a comparator and a summing amplifier which are the basis of library BIT options entitled "Comparator" and "Voltage Summing" techniques, respectively.

6.1.3   HYBRID SYSTEM

The Hybrid System considered for the survey was the Kaiser Electronics Display Processor, P/N CV-3916A. The processor is a current system in its final phase of development. Figure 6-3 is a diagram of a portion of the BIT circuitry. A built in test reference signal

- 96 -

PASS/FAIL

+VREF

−VREF

COMP

COMP

BIT
CIRCUITRY

XDEFL

YDEFL
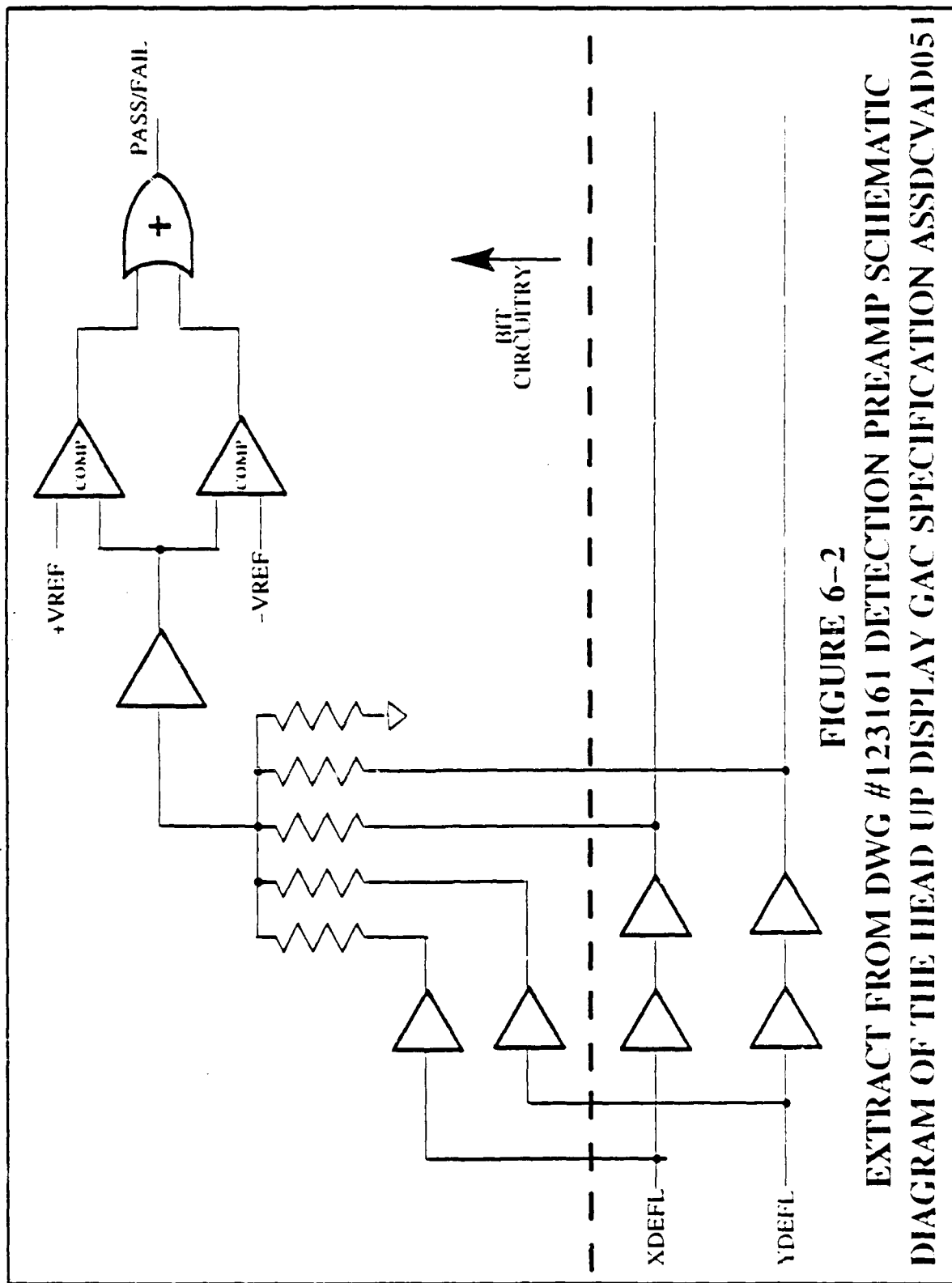
FIGURE 6-2

EXTRACT FROM DWG #123161 DETECTION PREAMP SCHEMATIC
DIAGRAM OF THE HEAD UP DISPLAY GAC SPECIFICATION ASSDCVAD051
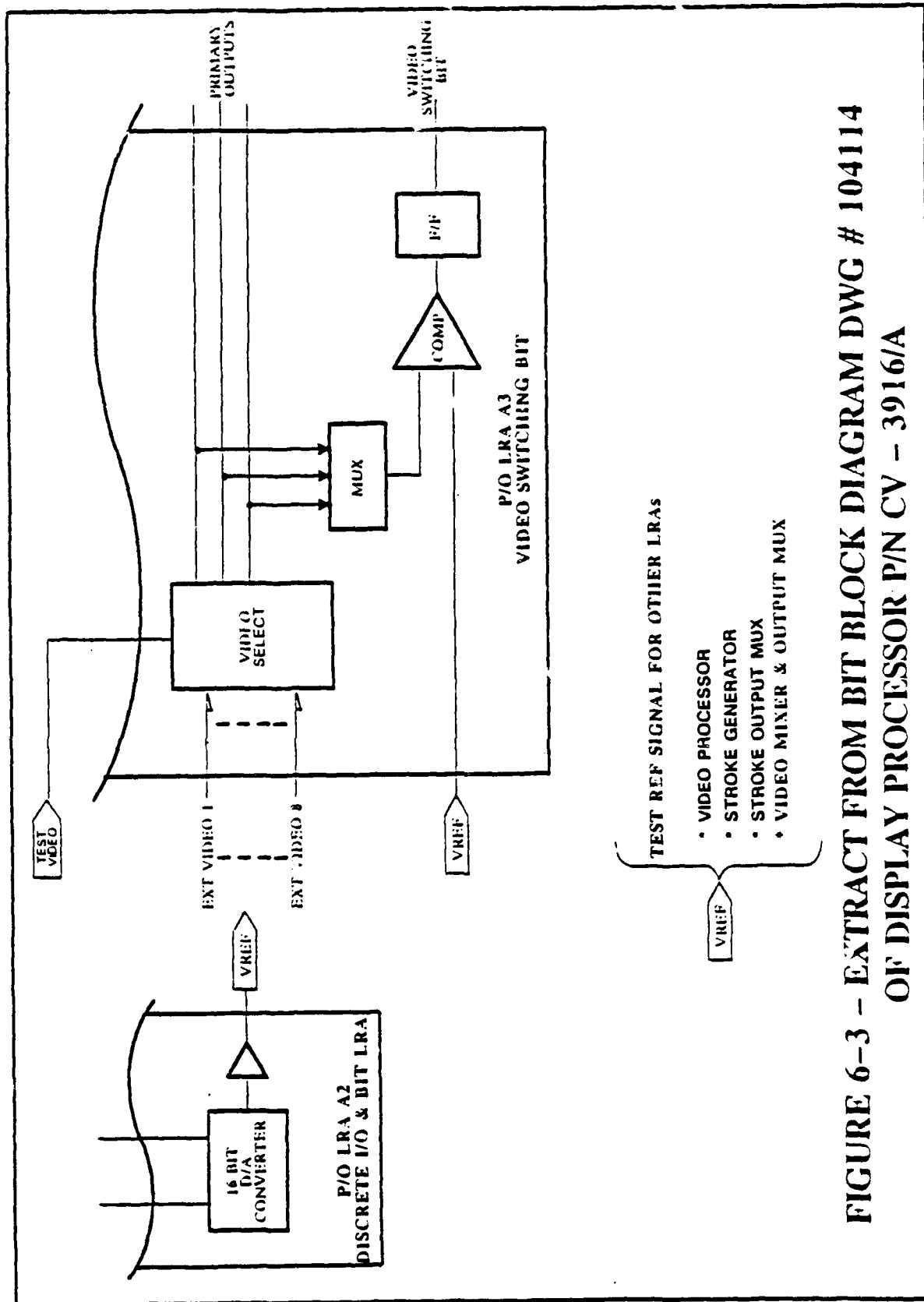
FIGURE 6-3 – EXTRACT FROM BIT BLOCK DIAGRAM DWG # 104114
OF DISPLAY PROCESSOR P/N CV – 3916/A

is generated by a D/A converter on LRA A2 and is routed to several other LRAs where the signal is fed into a comparator. The other input to the components is a primary output which has been forced by other comparators of the system or by a test signal source to provide a predetermined output. The reference voltage is adjusted to this predetermined output so that the comparator will detect primary output variations indicative of circuit failures. Although the BIT circuitry is distributed across several LRAs in this system, the above principle can be adapted to localization within a single LRM, thereby making it compatible with the BIT library ground rule that all BIT circuitry be self-contained within the LRM.

This system employs a D/A converter. If an A/D converter is added to the system, the output of the D/A can be fed to the input of the A/D and the comparison can be performed in digital form by the processor. The hybrid BIT technique "Analog Wrap Around" is based upon this and is considered to be a more general example of a hybrid BIT. However, the basic BIT scheme of the display processor has features that are also applicable to Analog BIT. An example is the use of a D/A to establish a test source or a test reference signal for the Analog BIT technique entitled "Comparator Technique".

6.2   LITERATURE SEARCH

In addition to analyzing current electronics equipment designs for BIT information, a Literature Search was also conducted. The search provided many papers on BIT from such representative sources as:

- Proceedings of IEEE International Test Conference

- Proceedings of International Conference of Computer Design

- Reports from Naval Ocean Systems Command

- Various Issues of IEEE Design and Test of Computers

The papers were reviewed to obtain detail design data for BIT techniques found in the BIT Equipment Survey and the papers also provided additional techniques for the BIT Library. A complete Bibliography is contained in Appendix B, Volume II, and applicable portions of the bibliography are listed at the end of the individual BIT techniques contained in the BIT Library, Volume II.

6.3 TESTABILITY BUS STANDARDS

Information was sought on the status of some of the proposed Testability Bus Standards. Among those considered were the VHSIC TM bus, the IEEE P1149 Testability Bus and the European originated Joint Test Action Group (JTAG) Boundary Scan scheme. As a result of our review it can be reported that recently (Spring of 88) JTAG joined forces with the IEEE group in an effort to achieve broader standardization. Scan is essentially the technique used on one of the equipments reviewed in the BIT survey, the JSTARS uAFP+ computer.

6.4 SMART BIT

Smart BIT is a program under development designed to reduce the number of false alarms, non-required maintenance actions and increase system readiness through a more effective application of BIT. It uses Artificial Intelligence techniques such as temporal knowledge and rule based systems along with local environmental factors and maintenance histories.

6.4.1 DESCRIPTION

Four Smart BIT techniques developed to date are:

* INTEGRATED BIT
* INFORMATION ENHANCED BIT
* IMPROVED DECISION BIT
* MAINTENANCE HISTORY BIT

The purpose of this technique is to reduce the false alarm rate compared to conventional BIT systems. Each approach focuses on a particular aspect of the problem. For example, both the Integrated BIT approach and the Information-Enhanced BIT approach attempt to improve BIT performance by providing a more global perspective on the BIT process. Improved Decision BIT rules attempt to improve BIT locally by replacing the simplified tests currently used with more robust tests. The Maintenance History BIT approach attempts to identify false alarms and intermittents by adding a historical perspective to the problem.

Since each approach highlights a particular aspect of BIT, they can be combined to produce a broad spectrum of improved BIT approaches. For example, at one extreme Information-enhanced BIT or Improved Decision BIT could be combined to improve BIT in a single Line Replaceable Module (LRM). At the other extreme, all four BIT approaches could be combined to produce a BIT system that encompasses much of an aircraft's avionics.

Figure 6-4 is a Level I block diagram describing how a Smart-BIT technique can be applied to CAD-BIT. The Smart-BIT Unit shown will be made up of electronic circuit components and may be a distinct module or combined with other BIT circuitry. These Smart-BIT Units will be located on PCBs, and therefore from the CAD-BIT point of view are merely other BIT components. Their use comprises additional BIT techniques. The overall CAD-BIT procedure remains as before with conventional techniques.
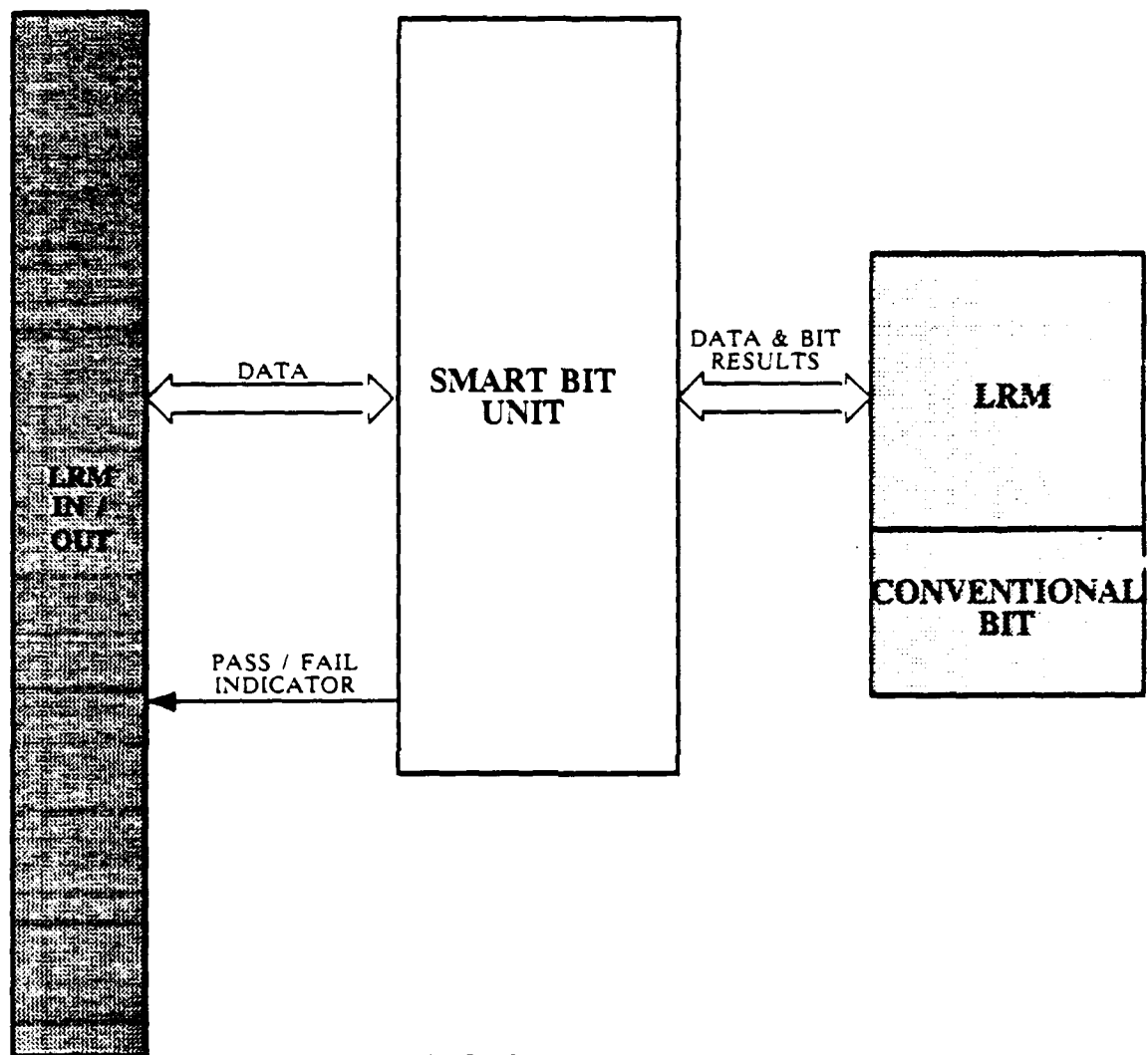
FIGURE 6-4

SMART BIT TECHNIQUE
LEVEL 1 BLOCK DIAGRAM

6.4.2 TECHNIQUES

(1) INTEGRATED BIT

The basic idea behind Integrated BIT is that BIT reports from several LRMs are passed to a higher-level BIT system for analysis. Results of the analysis are passed back to the lower-level LRMs. In a typical hierarchically Integrated BIT system, LRM BIT communicates with a subsystem BIT and that system communicates with yet a higher level one.

(2) INFORMATION ENHANCED BIT

In Information Enhanced BIT, BIT decisions are based not only on information internal to the LRM, but also on information provided by external sources. Typical information might include environmental stress, aircraft state and inputs and outputs.

(3) IMPROVED DECISION BIT

Improved Decision BIT reduces false alarms rates by providing a more robust diagnosis. This is accomplished by using parallel processes to cr ate a temporal monitoring structure with embedded rules. This replaces the conventional instantaneous testing with monitoring over time and use of external information in rule form.

(4) MAINTENANCE HISTORY BIT

The idea behind Maintenance History BIT is to make better use of the maintenance ' history of an LRM and the sequence of BIT reports during a mission. By analyzing such a BIT / Maintenance History for each unit, the real problems within the unit can be determined. This approach should be an effective way for identifying intermittent faults and separating them from false alarms. It requires rules interpreting a temporal data base.

6.4.3 SEQUENCE FLOW CHART DESCRIPTION

The BIT Sequence Flow Chart is illustrated in Figure 6-5. The steps are listed below.

1. Generate Conventional BIT results for the LRM.

2. Communicate results to the local Smart BIT unit temporal monitoring processes.

3. Generate Smart BIT analysis using appropriate mix of Smart BIT techniques.

4. If error is detected, set local FAIL indicator and pass information to the next higher level BIT.

5. If no error is detected, repeat sequence.
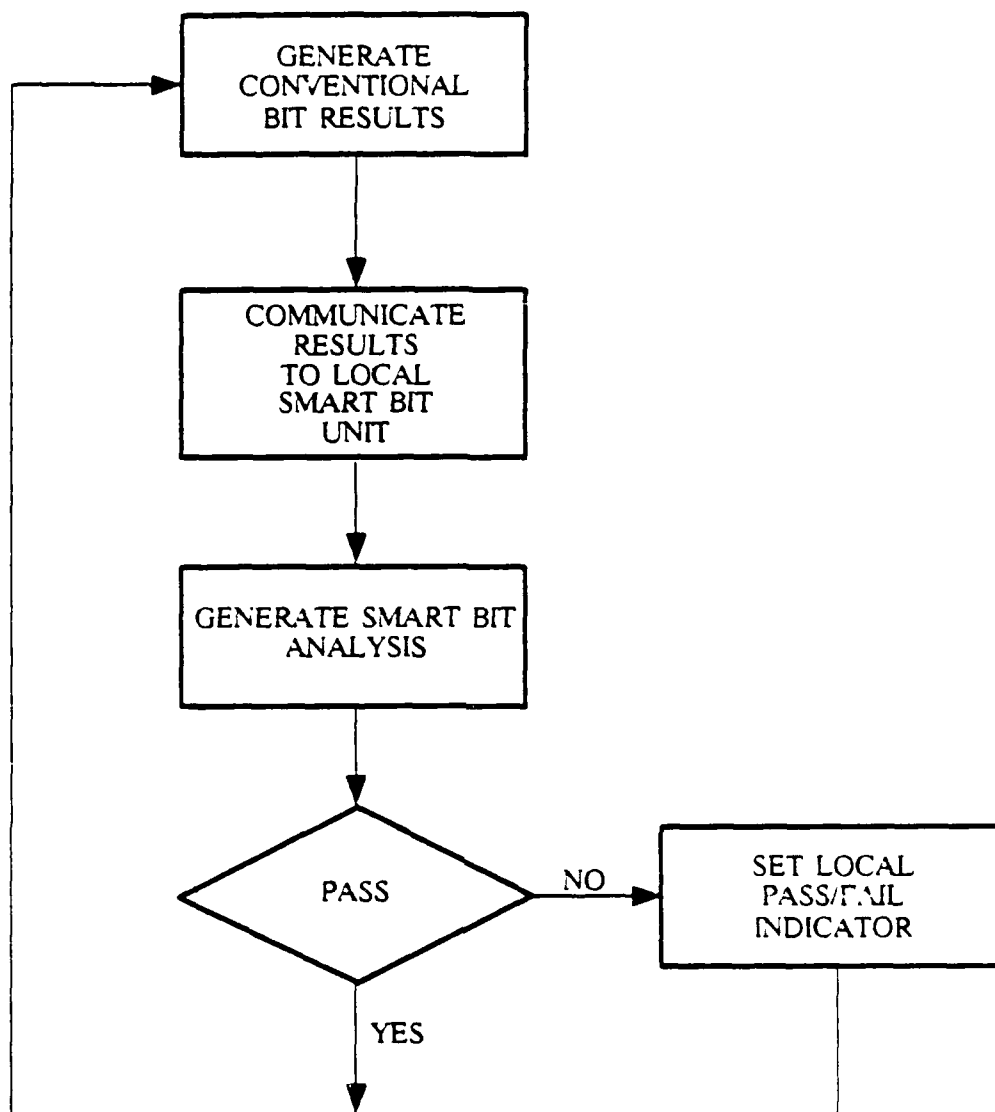
FIGURE 6-5

BIT SEQUENCE FLOW CHART

FOR SMART BIT TECHNIQUE

6.4.4 ADVANTAGES

The advantages of using Smart-BIT techniques are listed below.

1. Hierarchy allows easy extendibility and reduces overall system complexity.

2. Integrated BIT allows low level BIT reports to be analyzed by a higher authority.

3. BIT reports can be collected and compared over time, and used to filter the final BIT decision.

4. Having external information allows the BIT system to make decision it could not have previously made, making it more robust.

5. Diagnosis can be improved by the next level of hierarchy.

6. False alarm and intermittent patterns can be identified over time using maintenance history from several flights.

6.4.5 DISADVANTAGES

The disadvantages of using Smart-BIT techniques are listed below.

1. Delay in hierarchical elements can build up and become unacceptable.

2. Concurrent processing is required.

3. Maintenance data would have to be recorded if Maintenance History BIT is used.

4. Software and/or firmware is more complex.

6.4.6 ATTRIBUTES

The following attributes are characteristic of Smart-BIT techniques.

1. REAL ESTATE PENALTY: (See Note 1)

2. POWER PENALTY: (See Note 1)

3. RELIABILITY PENALTY: (See Note 1)

4. TIMING PENALTY: (See Note 1)

5. CONCURRENT

6. CONCEPTIONAL COMPLEXITY - Complex

7. HARDWARE/SOFTWARE/COMBO: Combination

8. TECHNOLOGY: (See Note 1)

9. IS BIT SELF TESTABLE: Yes

10. DESIGN COST - Cost of software and/or firmware considerable

11. STAND-ALONE (Self Contained BIT)

   No - Smart BIT is hierarchical and uses external information.

12. WEIGHT: See note 1.

   NOTE 1: Attribute unknown, pending further development of
   microprocessor technology.

6.4.7 BIBLIOGRAPHY

Additional information on Smart BIT is available from the following sources:

1. Dale W. Richards, "Smart BIT: A Plan for Intelligent Built-in-Test", IEEE NAECON, May., 1987, pp 1230 - 1233.

2. K. Haller, K. Anderson, J. Zbytniewski, L. Bagnall, "Smart BIT", RADC-TR-85-148, Aug., 1985.

3. K. Haller, J. Zbytniewski, K.Anderson, "Smart Built-in-Test (BIT)", IEEE Autotestcon, October, 1985, pp 140 - 147.

7.0 CAD SURVEY AND STANDARDS RECOMMENDATIONS

7.1 SURVEY OBJECTIVES

The objective of the CAD survey was to establish the state-of-the-art in CAD systems by researching the currently available CAD design systems and workstations being used for electrical and electronic design. The results of the survey were then used to make recomme ..ations for CAD-BIT Module (CBM) design standards.

The standards allow the CBM to be as transportable as possible on existing and future CAD workstations. Items identified as being important to the transportability issue include the workstation operating systems, program languages, and protocols. The protocol which is important to a successful CAD-BIT implementation is the graphical data transfer protocol. It will allow the transfer of BIT database graphical data between different CAD systems.

The survey and recommendation cited herein were made eighteen months prior to the writing of this Final CAD-BIT Report. Since that time, there has been no attempt to resurvey the CAD workstation situation as part of this contract. During this time, there has been no major reason to reverse the recommendations of standards relating to CAD-BIT. UNIX is as strong as ever, and if anything, has gained in popularity and strength. More is heard of Artificial Intelligence (AI), but CAD systems are still hosted on hardware systems where C is the predominant programming language. In the area of graphic interchange standards, IGES and EDIF are both competitors. IGES is the recommended standard, but this should still be reviewed prior to the next CAD-BIT phase.

7.2 CBM / CAD SURVEY ISSUES

The CAD market has historically been divided into two groups. The first consists of vendors who supplied full design systems (design software along with proprietary platforms and hardware). This group included some of the leaders in the CAD market such as Computervision, Daisy Systems, and Intergraph. The second group consists of those vendors who offer design software compatible with standard, "off-the-shelf" hardware or special proprietary hardware such as graphics boards or processors.

This survey revealed that most manufacturers of CAD systems are relying less on proprietary hardware and more on standard workstations. The most popular workstations are those made by Apollo Computer, Sun Microsystems, and DEC. Many vendors are also using IBM-PCs as a platform for their front-end schematic capture packages.

The following paragraphs discuss in detail the salient features of the CAD systems surveyed and their relationship to the CBM requirements definition.

7.3 OPERATING SYSTEMS

7.3.1 REQUIREMENTS

Choice of operating system will have a substantial effect on how the CBM will be developed. Both the design and its implementation will be affected by the variety of features supported by the operating system, including the nature of services supplied by it, and the method of communication between the CBM and the CAD system.

Among the operating system factors which affects the CBM, and therefore the CAD-BIT Software Specification, are the System File Structure and the available System Utilities. Since the CBM is not a stand-alone program, it is required that an operating system be selected prior to the development of the CAD-BIT Specification.

The file system and naming conventions of the operating system are important factors for CAD-BIT transportability. A consistent set of naming rules is required for the CBM to operate.

The CBM involves the interaction of many data files including standard text files and graphics files. The catalog structure and file definition can be seen in Figure 2-48. An operating system which can handle a hierarchical catalog structure is required.

By taking advantage of built-in utilities of today's operating system, the CBM will use existing features instead of recreating them. This will result in reduced development costs, higher reliability, and an overall lower software life-cycle cost of the CBM. However, the more one tries to take advantage of the utilities of a standard operating systems, the more one ties the CBM to that operating system, and the smaller the subset of CAD systems on which CAD-BIT can be applied. Therefore, a reasonable trade-off must be made in choosing an operating system which will enable CAD-BIT to be implemented over as wide a range of CAD workstations as is possible, but including systems powerful enough to handle the design of the Air Force's most sophisticated PCB designs.

As stated above, the use of an operating system's standard utilities will reduce the amount of code generated. Most operating systems contain sufficient utilities to meet the requirements of the CBM.

7.3.2 DISCUSSION

As a natural consequence of the trend towards standard hardware platforms cited above, there has been convergence to a smaller number of standard operating systems used on these platforms.

As can be seen in Figure 1-5, nearly all of the available CAD systems are based on only three operating systems: UNIX, MS-DOS, and VAX VMS. In general, the more powerful CAD systems use UNIX or VAX VMS, while MS-DOS is popular among the smaller CAD systems. In some cases, vendors use a combination of UNIX and MS-DOS, with the PC based MS-DOS portion used for schematic capture and simpler designs, but with more complex designs and Computer-Aided Engineering (CAE) functions performed on the higher-end system, UNIX.

Because of the predominance of these three operating systems in CAD applications, the remaining discussions will focus on these three operating systems.

(1)   VAX VMS

VMS is a proprietary operating system developed by Digital Equipment Corporation (DEC) and available only on their VAX line of computers. VMS was derived from RSX-11 and other operating systems that ran on the PDP-11 series. VMS has a rich array of utilities and features resulting from a long evolution on DEC equipment.

The overwhelming majority of installations of the VAX series run VMS; only a minority of sites are running either ULTRIX (the VAX implementation of UNIX produced by DEC) or some other implementation of UNIX for the VAX. Upon closer examination, a different picture emerges as regards the appropriate operating system for CAD-BIT even on VAX computers. Many of the VMS sites are commercial. Another large user category is educational time-sharing systems where the UNIX proportion is considerably higher. Among newly purchased VAX systems, the UNIX proportion is also much higher. This is true despite the fact that many sites were forced to choose VMS for compatibility with old VAX systems purchased before ULTRIX was available, and cannot migrate to UNIX due to the severe non-portability of VMS system services such as RMS.

(2)   MS-DOS

MS-DOS is a proprietary operating system that runs only on the IBM Personal Computer or compatible machines based on processors in the Intel 8086 family. MS-DOS originated as a CP/M work-alike for the 8086, and was known as SCP-DOS before Microsoft

bought the rights from Seattle Computer Products in 1980. At the time, Microsoft was also developing the "Xenix" version of UNIX, and deliberately made much of MS-DOS UNIX-like, with the intention of having the two systems tend toward convergence in the future.

MS-DOS is the operating system used on nearly all IBM and compatible Personal Computers (PCs) that have been purchased to date, and consequently there is a tremendous amount of PC software written to run under MS-DOS.

Unfortunately, most PCs lack the capacity to do any serious CAD work by themselves, and even the more powerful ones generally have to be networked to other processors and file systems. Fortunately, PCs can be networked to or downloaded from other processors and file systems, and may be very convenient for schematic capture. In these situations, the more powerful processor is probably running UNIX.

The net result is that the low end MS-DOS systems are not applicable to CAD-BIT.

(3)   UNIX

UNIX is a portable operating system, initially developed on a DEC PDP-7 computer at Bell Laboratories over a decade and a half ago. During most of the 1970's, UNIX enjoyed a long gestation period during which time it was enhanced, refined, revised, and rewritten several times both at Bell Labs and at several universities. UNIX began appearing in the computer market only within the past 5 or 6 years. Since then, however, a UNIX operating system has been implemented on nearly every major new computer.

Notwithstanding myriad innovations and extensive contributions to modern operating system design, its portability and hardware-independence are probably the most important features of UNIX. Even in cases where the ability to migrate to new hardware is unimportant, the user of UNIX still benefits from this portability. This is because virtually all of the applications software ever developed under UNIX, even on other hardware, is available to the user and does not have to be converted, emulated, modified, or rewritten. The only exception is for programs written in assembly language.

Thus, the wealth of utility software available under UNIX continues to grow with time. This is quite unlike the situation with any other operating system, where the useful life of the new hardware technology is often shorter than the development cycle for sophisticated software products to run on it. Standardized programming languages have not helped the non-portable operating systems much, largely because the services that software packages demand from the operating system are not standardized.

When a new software development project is UNIX-based, it immediately has access to all of the system services developed (more than a dozen years of evolution), plus all of the UNIX-based utility packages developed on all the different computers on which UNIX has run. Furthermore, the package can be used in conjunction with all of these utilities. In addition to these services and utilities, UNIX itself provides powerful features such as "shells" and "pipes", which can often make both the software development process and the use of software packages more productive.

7.2.3 RECOMMENDATION

The UNIX operating system is recommended as the one which best meets the requirements of CAD-BIT. It is the operating system which is used on most of the high end CAD systems, and it is the high end systems which are needed to meet the requirements of the more complex PCB designs required by the Air Force's PCB design efforts. Although the DEC VAX system using VMS also has sufficient computing power, it is a proprietary operating system and would be a very limiting factor in the availability of CAD-BIT.

By using standard workstations, UNIX has become the de facto operating system standard. All major workstation manufacturers offer UNIX or a UNIX-look-alike on their systems: Apollo provides Domain/IX, which is a combination of UNIX 4.2 and System V. DEC offers ULTRIX, which is based on 4.2 but includes System V enhancements; SUN furnishes SunOs, a combination of 4.2, 4.3, and System V; and IBM has AIX, an enhanced version of System V.

In choosing the UNIX operating system as the recommended operating system, the VAX VMS and the MS-DOS systems and the CAD software running under these systems will not be able to run CAD-BIT without additional work. The following paragraphs discuss the effect on CAD systems based upon these operating system with respect to CAD-BIT.

From a hardware point of view, the DEC VAX computer line may represent the largest potential of computing power for engineering workstations of any single manufacturer. There are two major considerations in reference to these DEC systems. One has to do with the operating system on which the CAD products run. The other has to do with a problem where the CAD software may now or in the future run under ULTRIX, but where the available hardware already is committed to VMS.

By the time CAD-BIT is in operation, more of the CAD software which is now written to run under VAX VMS can also be expected to run under ULTRIX. This is in line with a

general migration of most high end CAD systems to UNIX and the fact that once implemented under UNIX, they are more easily ported to other UNIX platforms.

The second issue is that even if all CAD applications could run under ULTRIX, the fact that VMS and ULTRIX cannot coexist on the same processor would present problems for those systems. The users which currently use VMS must continue to use it. One possible solution is the use of the DEC SHELL whereby the CAD product runs on VMS and CAD-BIT runs under the DEC SHELL.

In 1986 PC based CAD systems were estimated to capture 50% of the new workstations purchased in that year. At first glance, the loss of these workstations from CAD-BIT availability might seem a big loss.

The real effect of the loss to MS-DOS systems is probably not as large as the 50% figure seems. First of all many of those systems possessed limited capabilities and would not have been used for CAD-BIT applicable PCB designs for several reasons. The capabilities of these systems were limited in computer power, memory, and associated software and support. Required data transfer standards (IGES, EDIF, etc.) are much less likely to exist there than on the more powerful workstations. At the high end of the MS-DOS systems, one is more likely to find a UNIX capability. Another issue concerns the future convergence of the MS-DOS and UNIX operating systems.

7 4 PROGRAMMING LANGUAGES

CAD vendors' total software products generally involve programs running at the operating system level and the CAD application level. In addition, special sub-applications have been written in specialized languages. Such languages will not be included in this section, but will be included in section 7.6 along with the CAD database topics.

7.4.1 REQUIREMENTS

The programming language for CAD-BIT must be one which is commonly available on current state-of-the-art workstations. Programming language and the operating system interaction should also be considered since one can take advantage of the operating system's features to perform the implementation more efficiently.

7.4.2 DISCUSSION

Many of the CAD vendors surveyed provide the capability for users to create their own programs to access the design data bases. This is done by providing some form of linking

to their software subroutines. The trend is to provide a programming tool or language to allow access. By doing so, a user programmer can create custom software using calls to the existing software without the need for its source code or the need for a programming license. This capability is beyond the scope of the CBM, and would not yield a transportable module.

For transportability reasons the CBM cannot be written at the CAD application level. That would make it a custom program for each language used by CAD software vendors. Custom loading procedures would also be required. The CBM must operate at the operating system level, and there, the particular language in which the CAD application program is written, is not a factor. It is only necessary to choose a CBM programming language generally available on CAD hardware and operating systems.

7.4.3 RECOMMENDATION

The programming language recommended for the implementation of CAD-BIT is the C language, which is the same language in which most of the UNIX operating system is written. Use of C simplifies the interaction of CAD-BIT software with the operating system and provides convenient access to system services and utility programs.

However, it should be recognized that a UNIX-based CAD-BIT implementation could be done quite readily using another language. The operating system recommendation is in no way dependent upon the choice of C as a programming language. However, since all UNIX operating systems necessarily have C compilers and run-time libraries, the C language is therefore available on virtually all of the CAD systems surveyed. Furthermore, the semantics of a C language program are well-defined, and less likely to vary from one UNIX system to another.

Although Ada was one of the other languages considered for the implementation of CAD-BIT, the unavailability of validated compilers or run-time environments on most CAD systems made it inadvisable to consider Ada at this time.

FORTRAN 77 and Pascal were also considered. Although another procedural language could be used, calls to system services would be less convenient and less portable. Also, since not all compilers use calling sequences compatible with that used by the C compiler on a given UNIX system, the use of another language would make it difficult for the CAD-BIT module to take advantage of the large number of available tools already written in C. Furthermore, the semantics of FORTRAN and Pascal programs would be more likely to vary from one processor to another than that of a C program. Pascal presents

the additional problem that the syntax for interfacing separately compiled modules is nonstandard, and varies between compilers.

7.5 DATA TRANSFER PROTOCOL

7.5.1 REQUIREMENTS

As applied to the CAD-BIT program, the purpose of the interchange standard must first be determined before a logical choice of the standard can be made. The CBM is to be implemented on existing CAD systems and used to select BIT techniques and help the design engineer incorporate the selected BIT technique into the functional design. Electronic designs (electronic schematics) themselves have no requirement to be transferred. Neither do the CAD system schematic libraries nor their related data used by the CAD system utilities, simulators etc. Such libraries would not be compatible across systems.

The purpose for transferring graphical data between systems, especially upon delivery of CAD-BIT to a user, lies in the area of diagrams for tutorials and diagrams used to guide the designer in the implementation of the BIT circuitry into the functional design. It is most important that the graphics interchange standard be as universal as possible.

The need for data interchange standards became apparent in the late 1970's as organizations from industry and government began to acquire CAD systems from multiple sources or to deal with other organizations with different CAD systems. The problem has intensified as the number of different CAD systems has increased. Without a standard interchange capability, the number of translators for total communications between n systems is n(n-1). With a neutral format interchange standard the number of translators is reduced to 2n. Refer to Figure 7-1.

7.5.2 DISCUSSION

Other standards and specifications have been defined and tested. In the solid geometry area, these have included Applications Interface Specification (AIS), Experimental Boundary File (XBF), and CAM-I. A PDDI (Product Data Definition Interface) project has been started by the Air Force. Concepts from these have been incorporated into IGES.

In the international area, "Standard d' Exchange et de Transfert" (SET) was developed by Aerospatiale in France to overcome limitations with IGES and the very large file sizes resulting from IGES. It is based on IGES, but it uses a different file format resulting in reduced file sizes. SET appears to be limited in use to the European aerospace community.
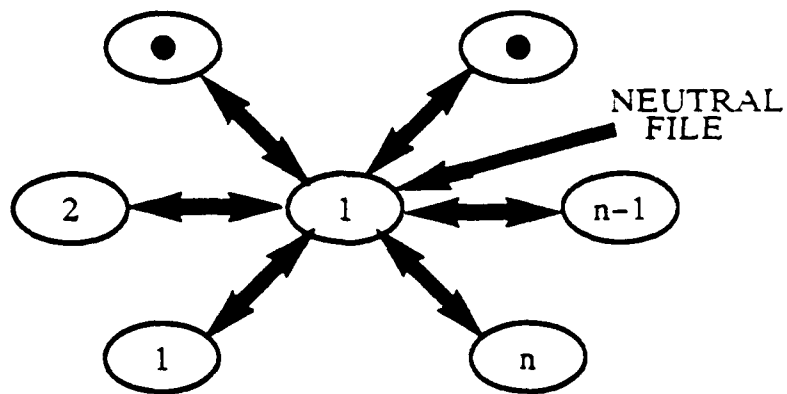
NEUTRAL FILE TRANSLATORS (2n)
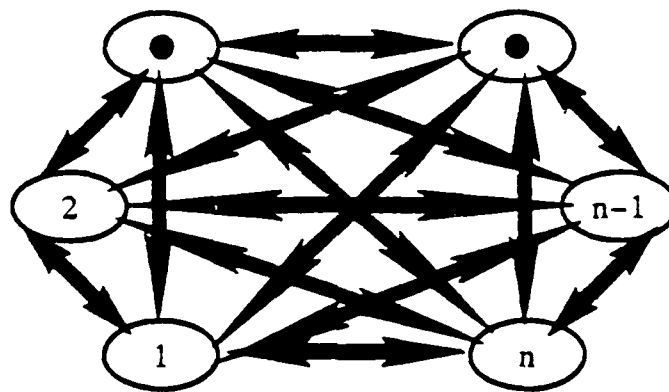
DIRECT TRANSLATORS (n(n-1))

FIGURE 7-1
THE DATA EXCHANGE SCENARIO

In 1983, within the International Standards Organization (ISO) a subcommittee (Subcommittee 4 within ISO Technical Committee 84 on Industrial Automation Systems) was organized to create an international standard. The proposed standard, The Standard for the Exchange of Product Model Data (STEP) is reviewing IGES, SET, PDDI, VDAFS, PDES, and CAD*I. A draft of the STEP standard is not expected before the end of 1987.

In the United States, the graphics interchange standards which have received the most attention and/or support have been IGES, Product Definition Exchange Specification (PDES), and EDIF. These are covered in more detail in the following paragraphs.

Since graphics transfer is the most important property of the CAD-BIT data transfer protocol, this eliminates the VHSIC Hardware Description Language (VHDL) which is excellent for transferring non-graphical electronic design data but lacks a graphics capability.

(1) IGES

The first version (Version 1.0) of the Initial Graphics Exchange Specification (IGES) was developed for the purpose of exchanging data between 1970's era CAD/CAM systems. These systems were drawing oriented and their data were primarily two or two and a half dimensional wire-frame in nature.

The specification was developed in response to an immediate need and consequently was put together rapidly in order to be of use in the short term. Unfortunately, in many cases this resulted in the specification being vague, ambiguous, and lacking in consistency. Additional flaws are its inefficient file structure, its inflexible data definition mechanisms, and its orientation toward graphical representation of a product's design.

Despite these shortcomings, the IGES has become the standard vehicle for the exchange of data for a variety of applications. The basic geometric entity set has been used to support the needs of a wide variety of users. Numerous vendors in the CAD area support IGES and their translators are in use with varying degrees of success. The simpler the database construction and the more basic the entity set, the greater the graphics transfer success. This is important for CAD-BIT where only simple graphics needs to be transferred and where the graphics to be transferred has not yet been created. The graphics entity set can be controlled resulting in far greater success. This is in fact the mechanism supported in the Computer-Aided Logistics Support (CALS) initiative where entity application subsets are being defined for the various CAD applications. For CALS, CAD users will be required to create their designs using only those entities specified in MIL-

D-28000, the Military Specification for Digital Representation for Communication of Product Data: IGES Application Subsets.

IGES has received much support from industry during its short existence. In the past year, IGES has also received strong support from the U. S. Navy and the Department of Defense.

A. NAVY CAD/CAM SPECIFICATION

IGES has been specified as one of the few non-negotiable mandatory items in the next phase of the Navy's CAD procurement program. The Navy CAD/CAM SECOND ACQUISITION TECHNICAL SPECIFICATION, Fourth Draft, Revision Q, dated 6 January 1987 specifies in paragraph 1.1.1 that "The Initial Graphics Exchange Specification (IGES) and Product Definition Exchange Specification (PDES) are public standards which are mandatory for the successful accomplishment of the Navy's mission. Any offerer which takes exception to the IGES or PDES requirements will be rejected without discussion". This specification calls for IGES, Version 3.0, but also requires IGES Version 4.0 preprocessor and postprocessor software within six months of the publication of the standard by the National Bureau of Standards.

Since this acquisition, represents the single largest planned CAD/CAM acquisition, it will be difficult for vendors to ignore these requirements. In the opinion of vendor representatives at the CAEP meeting to discuss this acquisition, they stated that their companies will do what is necessary to win a share of this future contract.

B. COMPUTER-AIDED ACQUISITION AND LOGISTICS SUPPORT

The Department of Defense's (DoD) Computer-aided Acquisition and Logistic Support (CALS) initiative is a cooperative DoD and industry strategy for transitioning paper-intensive design and logistic support processes to a highly automated and integrated mode of operation. Its objectives include integrating reliability and maintainability (R&M) engineering into industry automated design processes.

A primary CALS requirement is the development of a unified DoD / industry interface for the exchange of technical information in digital form. The unified interface will be achieved through a common core of functional and technical standards that will be used throughout the DoD in weapon system contracts. For Phase 1.0, three standardizing documents have been identified.

The first of the standardizing documents in the core group is MIL-STD-1840A (Automated Interchange of Technical Information). The purpose of this standard is to define

the mechanisms for transferring and storing in digital form, technical information necessary for the logistical support of a weapon system throughout its life cycle. In MIL-STD-1840A, MILITARY STANDARD DOD-D-(IGES) (now MIL-D-28000) and ANSI Y14.26M (IGES) are specified.

The second standardizing document in the core group is MILITARY STANDARD DOD-D-(IGES), the DoD Specification for the DIGITAL REPRESENTATION FOR COMMUNICATION OF PRODUCT DATA: APPLICATION SUBSETS. This specification defines applicable subsets of interest to the DoD. For each application subset, no ANSI Y14.26M (IGES) entities shall be used for information transfer or archival storage which are not enumerated in the defined subset.

One subset, Class III pertains to Electrical Printed Wiring Boards. The Electrical Printed Wiring Board subset addresses the representation and exchange of electrical/electronic products packaged on etched boards. Emphasis is on component descriptions, their placement, their conductivity, and routing of electrical paths. Both the physical view of the product (mechanical model or assembly drawing) and the logical view of the product (netlist or schematic drawing) are representable in the same file. Completeness and the functionality requirements of the received model for design manufacturing purposes are the basis of this subset.

The definition of this subset of IGES will remove much, if not all of the ambiguities previously associated with IGES transfers. Its inclusion in this core group of standards is expected to provide an incentive for additional industry support. It is also an indication that, although not perfect, IGES is the standard most suitable at this time and in the near future.

C. INDUSTRY SUPPORT

IGES is the only widely used and supported standard being used today. Many CAD systems which have working preprocessors and postprocessors into neutral formats support the IGES standard. This is not to say that they are totally satisfactory, but they are in use, serving a purpose for which they were intended, and are improving.

(2) PDES

The Product Definition Exchange Specification (PDES) is an interchange standard intended to fulfill a requirement beyond IGES's primarily graphics exchange. PDES is to contain further information such that the function of each entity or group of entities can

be determined. It's ultimate goal, therefore is to enable a computer system to understand and interpret the drawing file just as a trained engineer can understand the graphical paper drawing (which could have been generated via an IGES transfer). Such a standard therefore aims at communicating a complete product model with sufficient information to enable it to be interpreted by advanced CAD/CAM applications programs.

PDES is receiving government support from the same agencies that support IGES. Again, examples come from the Navy CAD/CAM Specification and CALS.

A. NAVY CAD/CAM SPECIFICATION

Support for PDES in the Navy CAD/CAM Specification follows the same general guidelines and requirements as stated above for IGES. These include the mandatory requirement to support PDES and the requirement that preprocessor and postprocessor software be provided within six months of the publication of the standard by the National Bureau of Standards.

B. COMPUTER-AIDED ACQUISITION AND LOGISTICS SUPPORT

PDES is called out in MIL-STD-1840A. However it is noted as under development. This development of this standard is considered as being too far in the future for CAD-BIT consideration.

C. INDUSTRY SUPPORT

Since the PDES Specification is not yet available, no meaningful support exists at this time.

(3) EDIF

The Electronic Design Interchange Format (EDIF) has come to the forefront as the CAE/CAD answer to IGES for the transfer of product data between vendors of CAD workstations for electrical and electronic applications. EDIF has been and is being developed for transmitting electronics design information. Its goal is to facilitate the information transfer among all systems involved in electronic design, test, and fabrication. For some developers of workstations and design systems, there is a pressure to make EDIF the cornerstone of their interface strategy. With the proliferation of design tools, companies are devoting more resources to interfaces. Making full use of EDIF would significantly increase the interface capabilities offered to customers.

A. EDIF GOVERNMENT AND INDUSTRY SUPPORT

EDIF, having been started by a group of CAE vendors, derives most of its support from these and other companies who have stated that they will support this standard. The actual support which this standard will receive remains to be seen since the standard is still in its early stages of definition and the number of preprocessors and postprocessors is not great and yet to be verified.

Daisy Systems, Mentor Graphics, and Cadnetix have all committed to full implementation of EDIF. Other manufacturers and suppliers of CAE/CAD systems are taking a wait-and-see attitude while keeping an eye on the development of EDIF. Many vendors prefer to use compilers that translate data directly into the format of the receiving system thereby reducing the possibility of data being lost in the format translation and also reducing the possibility of extraneous information being transferred.

B. NAVY CAD/CAM SPECIFICATION

The EDIF standard is not referenced in the Navy's CAD/CAM specification. This absence and the Navy's strong support for using IGES and PDES to address many of the applications beyond what IGES was intended, indicate that the Navy is not expecting to rely upon EDIF.

C. COMPUTER-AIDED ACQUISITION AND LOGISTICS SUPPORT

Like PDES, EDIF is called out in MIL-STD-1840A. However it is noted as under development. This standard may be developed enough to perform the graphics transfer required for CAD-BIT. It remains to be seen what true support will be provided by the CAE community in the long run.

(4) GRAPHICS STANDARDS

A great deal of emphasis has been placed on the use of software graphics standards. These are sets of graphic subroutines for input and output designed to be device independent. There appears to be more support for this software from the workstation manufacturers than from the CAE/CAD vendors.

Among the workstation manufacturers, Graphical Kernel System (GKS) and CORE have gained the greatest support. These graphics systems along with others are available as third party software to run on most workstations.

GKS is an application programmer's interface to graphics. The system is comprised of several subroutines used to generate hardware independent graphics. GKS is currently

under examination by ANSI as an industry-wide graphics standard. This is a cooperative effort to establish a method of transporting graphic applications from one computer to another. GKS is an effort to minimize future development costs for users. The advantage comes in purchasing GKS compatible software from multiple vendors and third party software suppliers with a minimum of support development. But there is a concern that GKS could be too slow for the high-performance graphic applications common to CAD.

The CORE system is another set of graphics programming subroutines that have been proposed as a standard. CORE has had some influence on the computer graphics industry particularly with Apollo Computers offering CORE and CORE-like systems on their workstations.

These standards are applicable to CAD vendors who require to port their products more easily between different hardware platforms. They do not apply to the transfer of CAD databases between different CAD systems.

7.5.3 RECOMMENDATION

IGES is recommended as the data interchange standard which best suits the requirements of CAD-BIT. IGES exists and has gone through several versions for which both pre-and post-processors have been developed and used successfully by numerous CAD vendors to translate design data between different CAD systems. In the evolution from the earliest versions to the present, IGES problems have been discovered and improvements made. These improvements apply to both the specification and the translators. IGES has been used for general purpose CAD system data transfer and has a range of applications in mechanical and electrical design.

7.6 CAD DATA BASE

The CAD databases include the parts data bases (schematics and other drawings), libraries, and other graphics and non-graphics supporting files.

7.6.1 REQUIREMENTS

The CAD database for CAD-BIT must allow the capability for the inclusion of user definable attributes or properties with numerical and text values during the design process. These must be able to be easily extracted programmatically. There must therefore be a data extraction capability as part of the CAD system. This capability will be required to evaluate the penalties (power, area, etc.) which result from the addition of the BIT circuitry.

- 121 -

7.6.2 DISCUSSION

The CAD parts databases are the "drawings" that the users generate. These drawings are typically more than just the presentation on the workstation display terminal or the hardcopy plot. Included in the drawing is other non-graphical information which is used by the internal software to interpret the meaning of the visual data. This data may include engineering data (electrical connectivity data or simulation input data), process control data, manufacturing data, or data for other automated applications.

Each CAD system has its own way of storing and interpreting this data. The particular way this is done is irrelevant to CAD-BIT. The schematic design should include the data which is required for the BIT circuitry insertion and the BIT penalty evaluation. This is related to the data extraction capability of the system.

A second matter of importance is the capability to "PUT" and to "GET" graphical data from different CAD systems in the form of IGES, EDIF, etc. neutral file databases. These will be required for the transfer of tutorial information. In a more advanced version of CAD-BIT than is within the scope of this project, it could include "intelligent tutorial" data which could automate some aspects of the BIT circuitry interconnection process. That automated process would normally not be transportable, but if done on the neutral database it would be feasible. It would require preprocessors and postprocessors with a speed and reliability which surpasses anything in existence today.

The CAD library situation is similar to the CAD part database situation. Each CAD system's libraries are in a proprietary format and contain some common and unique data. The unique data has to do with the automated software applications on the system. No standards pertaining to libraries exist nor are CAD system libraries compatible.

7.6.3 CAD DATA EXTRACTION

Data Extraction is the ability to programmatically extract data from the CAD design data base. This data is information that is obtained from the "drawing", or may be data internal to the drawing or library elements. For example, one may observe an electrical schematic and directly observe electrical connectivity between two points. One may also find non-graphical data in the data base which is not normally visible, but which can be extracted. This may include various attributes or attribute values referring to the particular design or data (timing, simulation, thermal, etc.) about the electrical component.

Data extraction techniques vary from system to system. They generally are of the two types discussed in the paragraphs below. There are no standards for CAD system data

extraction at this time. Data extraction requirements for the CBM will have to be handled by the users based upon the CBM software specification. Such data extractions will be required for the BIT penalty evaluation. Many of the CAD vendors provide for users to create their own programs to access the design data bases. This is done by providing some form of linking to their software subroutines. The trend is to provide a programming tool or very high order language to allow easy access to database parameters. By providing this tool, a user need not be a "programmer" to extract needed data.

(1)   Extraction Via Vendor Supplied Data Extraction Languages

Data extraction capabilities are often in the form of vendor specific higher order languages in which the user specifies the data element types to be extracted using keywords, data extract procedures, and the format of the generated report. Such data extract capabilities may vary from bill of materials to netlist type and are user definable in content and form. These data extract languages are intended for non-programmers and are generally easy to write.

The procedures themselves are supplied by the CAD vendors and are often undocumented or difficult for users to write. Therefore if the user wishes to extract data which is outside that planned by the vendor, difficulties may exist.

(2)   Extraction Via Higher Order Languages

Another method of data extract involves the the use of higher order languages (often the CAD application source language) which link directly with the vendors subroutines. The user must often be familiar with the CAD data base structure in order to use these effectively. Vendors sometimes supply a Macro language and compiler to make the above process easier.

7.6.4 RECOMMENDATION

There is no specific recommendation relating to these issues for CAD-BIT. The CAD databases are what they are. On a reasonably good CAD system the database will be adequate for CAD-BIT to function. A more crucial issue will be the friendliness and effectiveness of the CAD system, but that is outside the scope of CAD-BIT and this report.

7.7 CBM / CAD ENVIRONMENT

The tools discussed are generally available on CAD systems to different degrees. There is little or no standardization in these areas, but it will be important for these to be taken into consideration in the design and implementation of the CAD-BIT specification.

7.7.1 EXECUTE FILES

Execute Files contain commands or instructions to be executed sequentially or according to the result of expressions and branching or looping statements. For example, at the operating system level in UNIX, these may be shell scripts. At the CAD application level they may be special commands to process graphics or applications commands. For example on a Computervision system they are known as "Execute Files" and on a Mentor Graphics system they are "DO" commands.

At the operating system level for CAD-BIT it is assumed that execute files such as shell scripts will exist. This will enable the system functions to be called along with CBM shell scripts to be used together to easily implement various CAD-BIT functions.

At the CAD application level, it is assumed that an "execute type function" will be available. Specifically, it is anticipated that during the BIT insertion phase on the CAD system, commands will be generated at the operating system level which will be implemented at the CAD level by the calling of a fixed name execute file. In the event that a particular system does not have any execute type command capability it will require the user to issue the specific command in its entirety rather than call the execute function. The capabilities of CAD-BIT on such a system will be the same, but the user friendliness will be reduced.

7.7.2 PROGRAMMING ENVIRONMENTS

CAD systems generally consist of the CAD Application Program(s) running as applications within the host operating system. The user of CAD-BIT, assumed to be an electronic design engineer, may escape from the CAD application program without terminating either the CAD application or the circuit being designed. Once in the operating system, utilities or other programs may be utilized. Thus there are two environments available to the designer: (1) the operating system, and (2) the CAD environment. This is necessary for CAD-BIT since the designer will move back and forth between the CAD environment for circuit design and BIT circuit insertion and the operating system environment for various CAD-BIT non-graphic functions.

The changing of environments is performed differently on each CAD system. The method may involve issuing a command to change environments, or may be performed by simply moving a mouse cursor to a different window. In either case, the operation cannot be performed in a transportable manner on different CAD systems since that

operation is embedded in the CAD application program. The scenario for changing environments, therefore, needs to be whatever method is available on each CAD system.

Because of the reasons cited above, the automatic linking between environments for various CAD-BIT functions cannot be implemented in a transportable manner. Such a feature would require a major software effort for each CAD system on which CAD-BIT is used.

7.8 WINDOWS

Most modern CAD systems feature some sort of "windows". For maximum friendliness, the CAD-BIT module must be capable of operating via windows while the underlying CAD system is in use.

As more CAD systems run on the standard hardware and software platforms, it appears that these systems will have a windowing capability which will enhance the addition of added software such as CAD-BIT.

8.0 AUTOMATED PROCEDURE EVALUATION

The CAD-BIT demonstration provided a test bed for the development of a user-friendly automated procedure. The evaluation of this automated procedure is summarized below with respect to feasibility, practicality, transportability, extensibility and cost-effectiveness to implement.

8.1 IMPLEMENTATION COST EFFECTIVENESS

The tasks required to implement CAD-BIT on a new UNIX system are included in a preceding section. All tasks listed are relatively small, and do not require a skill level above that normally required to support a CAD system. The menu customization task is the largest of these. This task will take a minimum of several weeks to implement depending upon the host CAD system menu capabilities, tools, and user-friendliness.

8.2 FEASIBILITY

The feasibility for the CAD-BIT procedure to operate on a CAD system was demonstrated. It was shown that the procedure could coexist with the CAD application by having the CBM operate at the Operating System level and interact effectively with the CAD application software. A transportable CAD-BIT procedure operating at the CAD level is not possible. This is because of the different ways user added software is interfaced to CAD application software, the way CAD menus are embedded in the CAD application, and because proprietary software tools are required to compile, link, and load each vendors CAD application program.

8.3 PRACTICALITY

The CAD-BIT procedure developed was demonstrated to be practical. A user-friendly procedure can be created and easily interfaced to the CAD application software, with the CAD-BIT software remaining external to the CAD software. Changes in one do not affect the operation of the other. The BIT data base is transported between CAD systems via standard ASCII files. The graphics files in Initial Graphics Exchange Specification (IGES) are also ASCII. The BIT data base structure is extensible in all respects. Any number of techniques may be added, and additional BIT suitability, selection, penalty, etc. parameters can be added without practical limits subject to memory limitations.

8.4 TRANSPORTABILITY

The system is easily transportable. Implementation tasks include the following:

- 126 -

- Loading the software and BIT data base into a specified directory structure via a standard Operating System command
- Compiling, linking, and loading via a C-shell script
- IGES conversions
- Creation of CAD menus
- Alteration of several of the data base files for host system command syntax, and directory and file names.

8.5 DATA BASE EXTENSIBILITY

Extensibility is provided through the data base design, where any number of additional BIT Technique data sets and any number of BIT suitability, selection, and penalty parameters may be added. These are often added using utilities to facilitate the process.

8.5.1 ADDING TECHNIQUES

New techniques may be implemented by the addition of that technique's data and figure files in its own subdirectory under the technique name. All files for any technique reside in a single directory. Figure 2-48 shows the CAD-BIT data base structure. The boxes enclosed with heavy dashed lines illustrate the files which must be added when implementing a new BIT technique. The figure also shows how they are added.

The Technique list must be updated to include the name of the added technique, the new technique sub-directory created, and the technique-file attribute file created. These functions are facilitated through the use of the ADD TECHNIQUE UTILITY.

The program used for the selection of the default technique components, their quantity, and the calculation of BIT penalties associated with the technique must be created. These are created interactively through the use of the FIX-PENALTY UTILITY. This utility generates C language source code, and after being compiled and loaded, it generates the final program for the added technique.

The technique circuit-menu list is generated via the GENERATE-CIRCUIT UTILITY. It generates the menu used for BIT insertion and creates additions to the parts-master file if new circuit elements are required.

8.5.2 ADDING SUITABILITY PARAMETERS

Additional BIT suitability parameters may be included by adding a line entry into the Suitability Attribute List section of the Profile Template File. The line entry must include the suitability attribute name, default answer, allowable answers, and suitability question. The suitability attribute help file must be created in addition. Figure 8-1 (ADDING
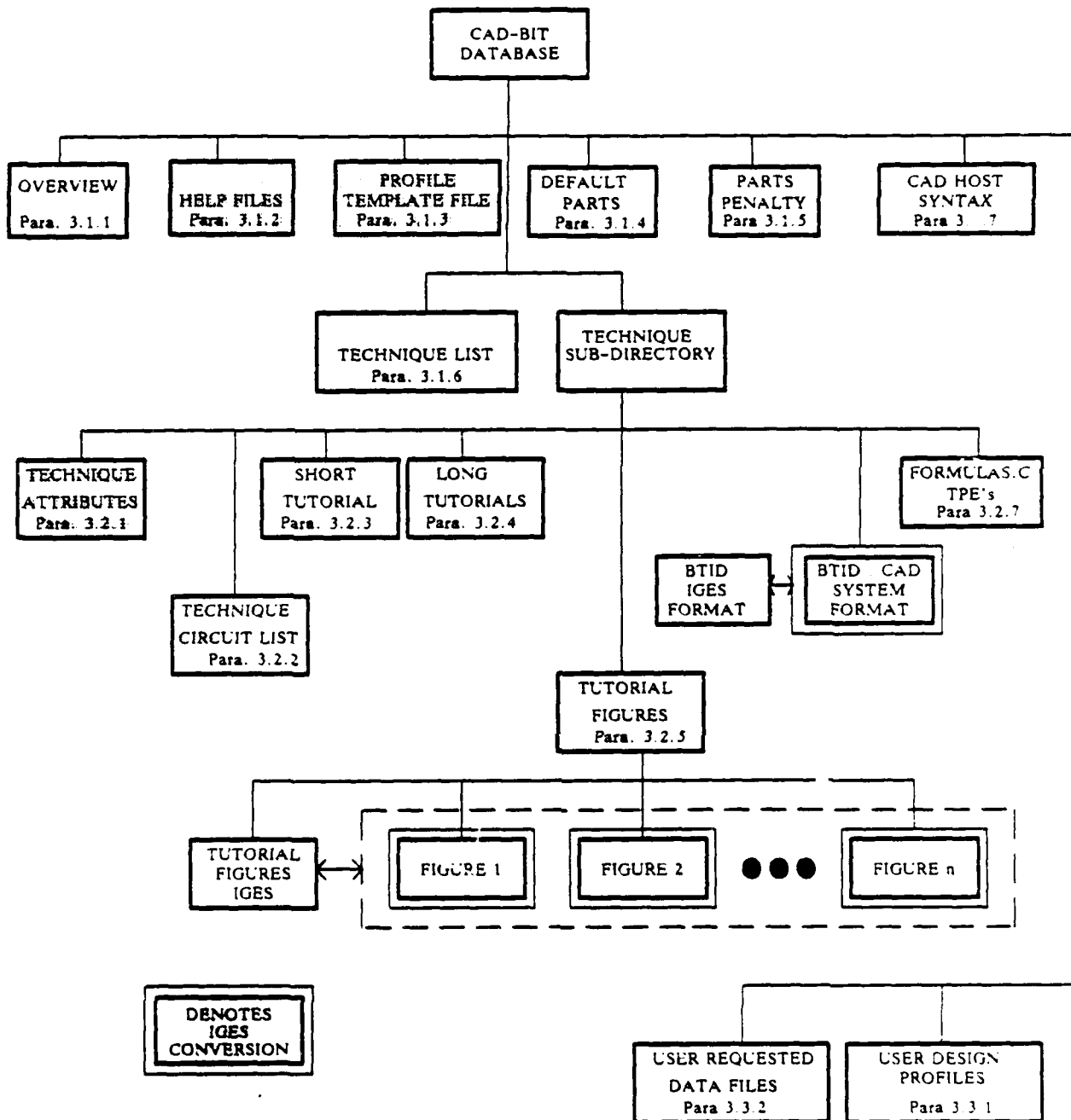
Figure 8-1

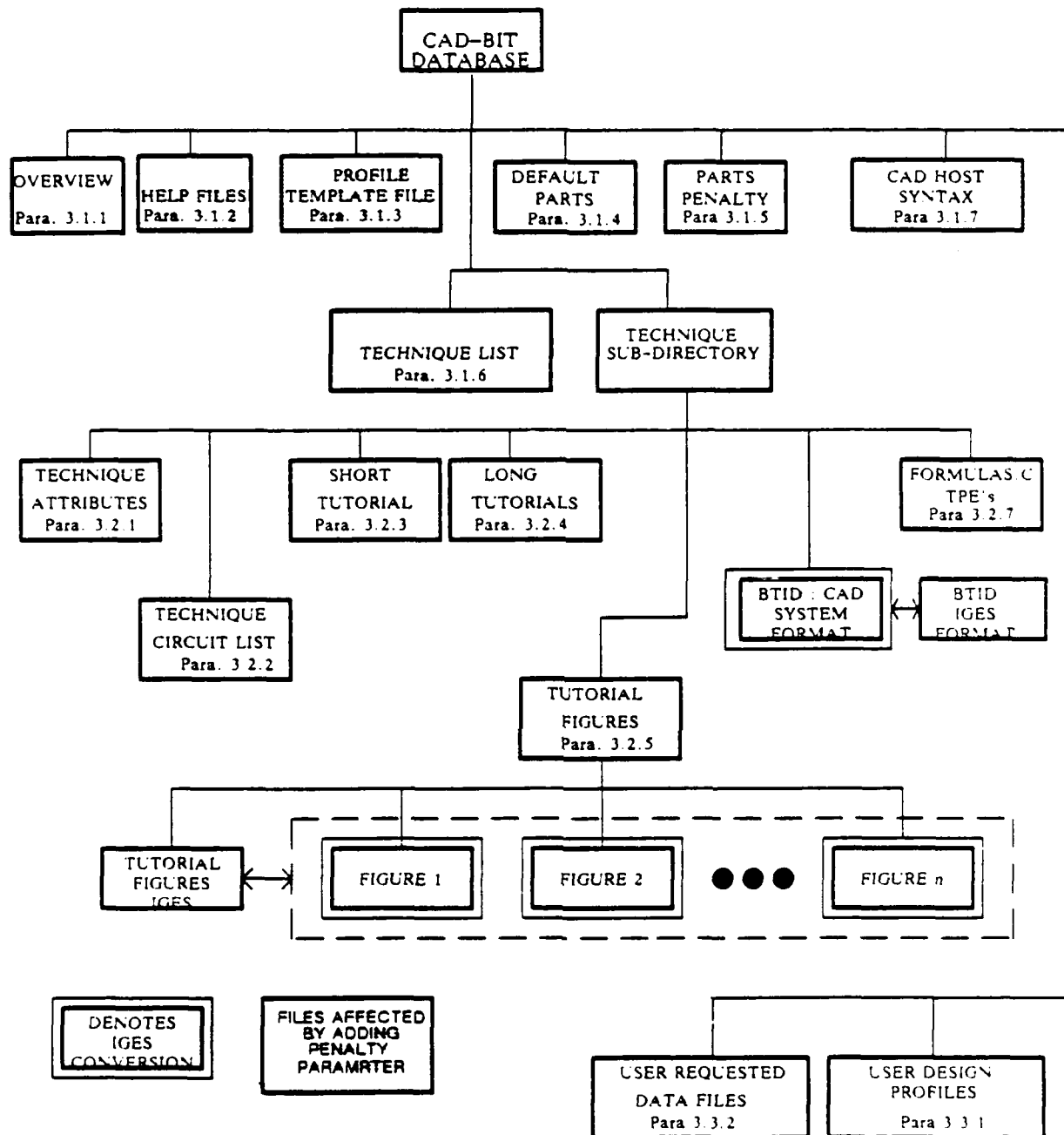**EXTENSIBILITY : ADDING SUITABILITY PARAMETERS**

SUITABILITY PARAMETERS) illustrates the files affected by the addition of suitability parameters. The incorporation of new suitability attributes may also affect old user design profiles if they need to be re-run after the attribute additions, and if the new attributes are to be considered.

8.5.3 ADDING PENALTY PARAMETERS

Additional BIT penalty parameters may be included by adding a line entry into the Profile Template File PPL. The line entry must include the penalty parameter name, default penalty weighting factor, and penalty parameter weighting factor question. In addition, the suitability attribute help file must be created. Figure 8-2 (ADDING PENALTY PARAMETERS) shows the files affected.

8.5.4 UTILITY FUNCTIONS

Utility functions are provided for various functions associated with extensibility. These include the ADD-TECHNIQUE, FIX-PENALTY, and GENERATE-CIRCUIT utilities discussed in the paragraphs above on extensibility.

Figure 8-2

EXTENSIBILITY : ADDING PENALTY PARAMETERS

APPENDIX A
BIT LIBRARY EXAMPLE

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: SHORT TUTORIAL

SUBCATEGORY: DESCRIPTION OF BIT TECHNIQUE

DATA TYPE: TEXT ☒   LIST ☐   TABLE ☐   GRAPHIC ☐   EQUATIONS ☐

DATA:

SHORT TUTORIAL
FOR
ON-BOARD ROM

On-Board Read Only Memory (ROM) Self Test is non-concurrent, mostly hardware and firmware, Built-In-Test (BIT) technique which consists of applying test patterns that are stored in an on board ROM to a Circuit Under Test (CUT) and then comparing the CUT's response to what is expected, resulting in a go – no/go output signal. Although the number of test patterns required to exhaustively test a function is proportional to the cube of the number of gates, this technique has some potential in that each test pattern can be individually and selectively determined, thereby, maximizing the percentage of fault detection to the test pattern ratio.

LIBRARY ELEMENT DATA
SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: SHORT TUTORIAL

SUBCATEGORY: 1. LEVEL I BLOCK DIAGRAM.
2. BIT SEQUENCE FLOW CHART.

DATA TYPE: TEXT ☐ LIST ☐ TABLE ☐ GRAPHIC ☒ EQUATIONS ☐

DATA:

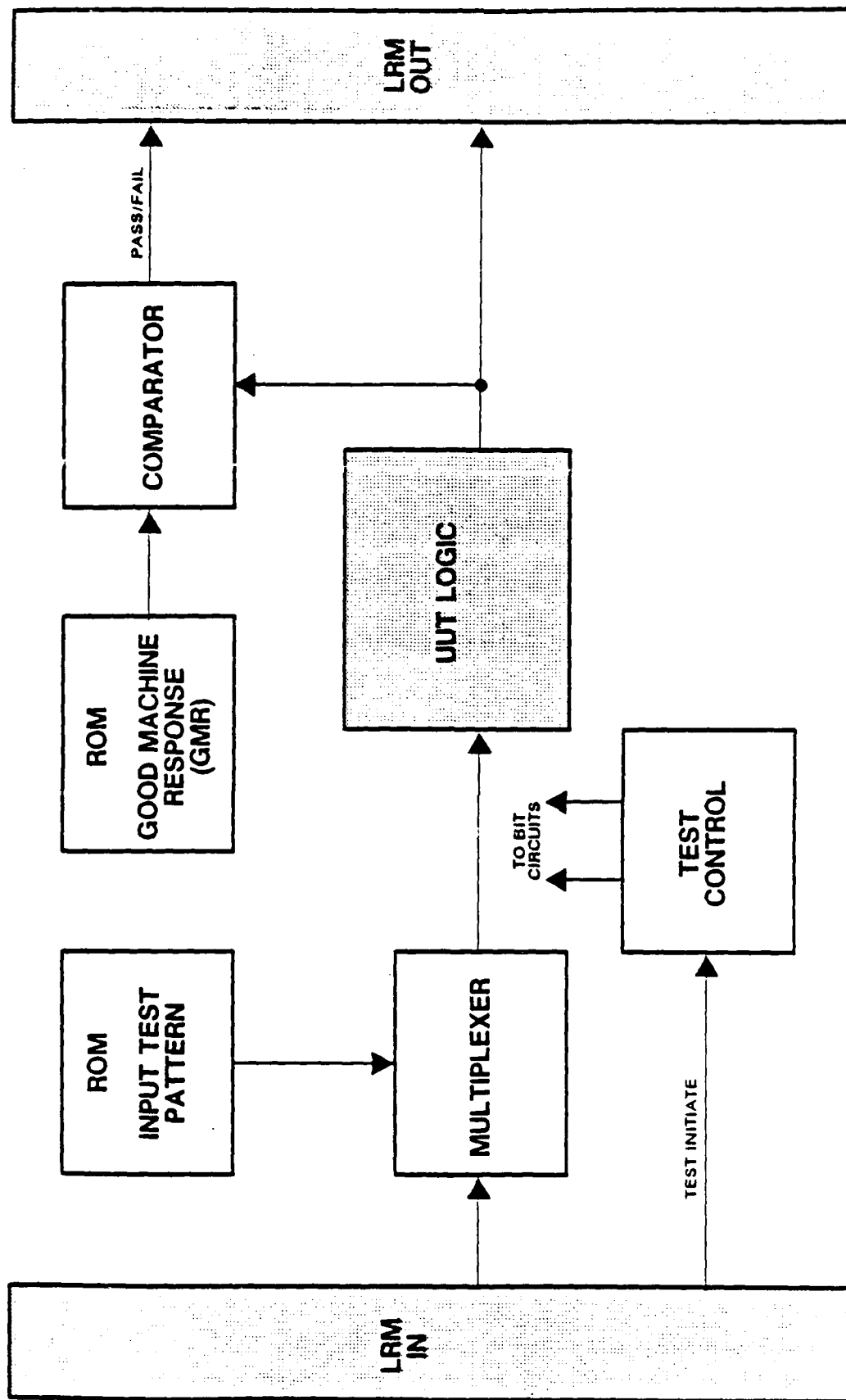SUBCATEGORY 1: SEE FIGURE 1

SUBCATEGORY 2: SEE FIGURE 2

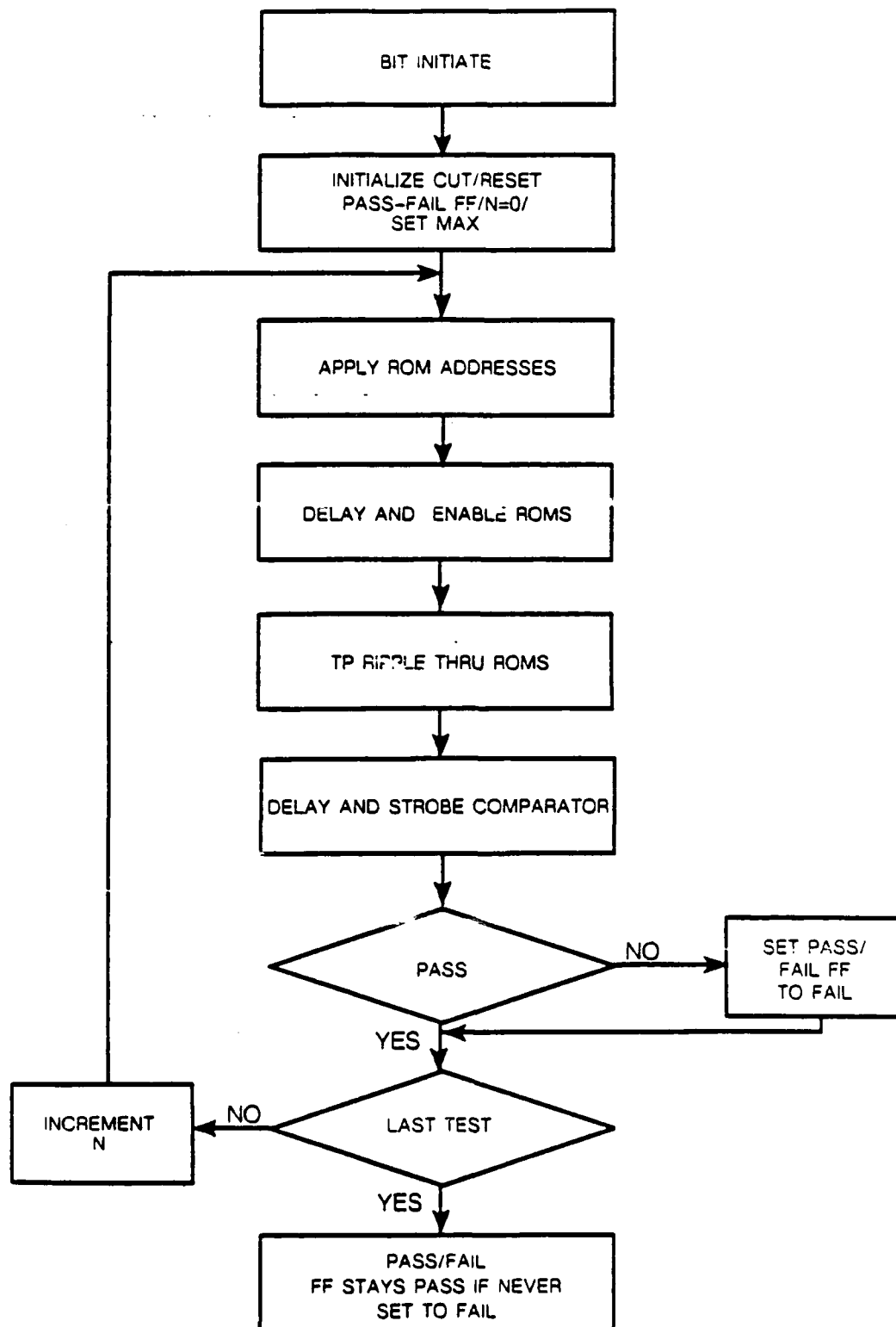FIGURE 1 – LEVEL I BLOCK DIAGRAM FOR ON-BOARD ROM

A-3

FIGURE 2 – BIT SEQUENCE FLOW CHART
FOR ON-BOARD ROM

LIBRARY ELEMENT DATA
SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: LONG TUTORIAL   **PAGE** 1 of 14

SUBCATEGORY: BIT SEQUENCE FLOW CHART DESCRIPTION

DATA TYPE: TEXT ☐   LIST ☒   TABLE ☐   GRAPHIC ☐   EQUATIO⹁ ☐

DATA:

BIT SEQUENCE FLOW CHART DESCRIPTION
FOR
ON-BOARD ROM

1. A positive pulse "Test Initiate" signal is input to test control logic to begin test.

2. The test begins as follows:

   - "BIT Mode" signal from control logic to multiplexer is activated

   - Normal inputs to CUT multiplexed out

   - Test Patterns (TP) from TP ROM input to CUT enable

   - All resettable logic of CUT reset

   - ROM address counter in control logic reset to zero

   - "Pass/Fail" Flip-Flop (FF) in comparator logic block reset to "Pass"

3. The system clock, while in BIT mode, increments the control logic counter which addresses the TP & Good Machine Response (GMR) ROMs simultaneously.

4. After a delay sufficient to fully establish the addressing in the step above, both the TP & GMR ROMs are enabled.

5. The TP ripples through the CUT. To gain control of the CUT clock, each TP will have both high and a low on the clock line which may come from the TP ROM.

6. After enough delay for a good machine to establish a GMR at the CUT's outputs, the comparator is enabled.

7. A good machine at this time will have the GMR pattern identically compare with the CUT outputs. If not, the Pass/Fail FF will be set to "Fail" and wi" remain "Fail" until BIT is re-initiated.

8. If the address to the ROMs is the last address, then "End Of Test" control logic signal goes low. The moment the enable comparator signal goes HI during this last TP sequence, the BIT mode FF is reset and the system is out of BIT mode. The Pass/Fail FF will remain set to "Pass" if during the test it was never set to "Fail".

9. If not the last ROM address, go back to step 3.

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: LONG TUTORIAL

SUBCATEGORY: BIT TECHNIQUE ADVANTAGES

DATA TYPE: TEXT ☐   LIST ☒   TABLE ☐   GRAPHIC ☐   EQUATIONS ☐

DATA:

ON-BOARD ROM BIT ADVANTAGES

1. An understanding of the CUT can lead to a substantial percentage of fault detected with a few, determined test patterns.

2. A CUT with much sequential logic requires specific "Pairs" of test patterns applied in sequence. Although, this presents a problem with Random Test Pattern Application, storing the test patterns in ROM so that they indeed do occur in pairs is done without difficulty with the On-Board ROM Method.

3. On-Board ROM Test Generation becomes competitive when compared to random pattern generation as the number of CUT inputs become large and/or number of patterns required becomes small. This is best understood by considering that the total number of binary patterns possible for a CUT with n inputs is 2^n. If n=16; 2^n = 65,536. If n=20; 2^n = 1,048,576. If n=24; 2^n = 16,777,216. Consider a hypothetical 24 input CUT that can be adequately tested with 2,000 deterministic patterns. Most of the Test Pattern Generator (TPG) hardware required using On-Board ROM Method are cascaded, 2K by 8 ROMs as compared to 3 cascaded, 8-Bit shift registers plus 2 Quad Exclusive Or Packages. But the real savings is test time. To be absolutely sure of providing all 2,000 test patterns one must cycle through 16,777,215 possible test patterns when using random pattern generator.

4. The control logic for the On-Board ROM Test is simple when compared to the Random Test Pattern Generation method which requires loading seed patterns and special test sequencing.

5. Read control logic and address and data buses may possibly be shared between test and function purposes.

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM BIT

SUBCATEGORY: BIT TECHNIQUE DISADVANTAGES

DATA TYPE: TEXT ☐   LIST ☒   TABLE ☐   GRAPHIC ☐   EQUATIONS ☐

DATA:

ON-BOARD ROM BIT
DISADVANTAGES

1. With the growing complexity of electronic circuitry being implemented on Line Replaceable Modules (LRM) of today, it is becoming more and more difficult for a test engineer to understand what he is testing, especially when under pressure to establish the test plan quickly. Without a true understanding of what is to be tested, it is nearly impossible to effectively and efficiently determine the test patterns that are necessary.

2. When the number of test patterns required to obtain adequate fault coverage is large and/or the number of CUT inputs is small or can be partitioned into a few small number of input groups, then the real estate required for the On-Board ROM Method becomes excessive when compared to the random pattern generation method.

3. Memory elements in general are not as reliable as random logic microelectronic devices.

4. Circuit design changes often require reprogramming the ROMs.

5. If the number of bus lines required to address the ROMs are excessive and/or the distance between the TP ROMs and the control logic, or between the GMR ROMs and the control logic is substantial, then Printed Circuit Board (PCB) real estate consumed is excessive and costly.

6. Memory allocated to either store test patterns or GMRs can never serve both test and function purposes as can shift registers used in Built in Logic Block Observers (BILBO) for example.

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: LONG TUTORIAL   PAGE 4 of 14

SUBCATEGORY: BIT TECHNIQUE ATTRIBUTES

DATA TYPE: TEXT ☐   LIST ☒   TABLE ☐   GRAPHIC ☐   EQUATIONS ☐

DATA:

ON-BOARD ROM BIT ATTRIBUTES

1. REAL ESTATE PENALTY

    * Increases with CUT complexity

    * ROMs - Number of test patterns is approximately the cube of the number of gates for combinational. FFs increase the number even further

    * Control - Approximately 11 chips for this example. Number of counter chips increases with number of test patterns

    * Multiplexer - Number multiplexer chips equals number input lines divided by number of lines switched by multiplexer chip

    * Comparator - Number comparator chips equals (number output lines) divided by number of lines compared by chip

    * Land real estate depends on layout

2. POWER PENALTY

    * Roughly proportional to real estate penalty example:
      Power Penalty equals Percent Real Estate Penalty multiplied by CUT Normal power.

            - Exceptions (some ROMS have power down mode)

            - Switch Technology (use Metal Oxide Semiconductors (MOS) ROMS for higher density)

3. RELIABILITY PENALTY

    * Proportional to Real Estate Penalty if similar technology is used for Built in Test Equipment (BITE) as for CUT

    * May have to distinguish BITE failures that only effect BITE vs BITE failures that effect CUT

    * Computer Aided Design (CAD) System may have software package for reliability calculation

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: LONG TUTORIAL	PAGE 5 of 14

SUBCATEGORY: BIT TECHNIQUE ATTRIBUTES

DATA TYPE: TEXT ☐   LIST ☒   TABLE ☐   GRAPHIC ☐   EQUATIONS ☐

DATA:

ON-BOARD ROM BIT ATTRIBUTES (CONT)

4. TIMING PENALTY

   * Test Time Duration - Number of Test Patterns multiplied by Pattern Application Period

   * Circuit throughput Delay - Additional delays of Multiplexers

5. NON-CONCURRENT

6. CONCEPTUAL COMPLEXITY

   * Straight Forward

7. HARDWARE/SOFTWARE

   * Test Patterns in Firmware

8. TECHNOLOGY

   * All current digital technologies

   * May use higher density technologies for ROM to reduce real estate penalty. (May need MOS-Transistor Transistor Logic (TTL) converters)

9. IS BITE SELF TESTABLE?

   * Can do check sum on ROMs (add hardware)

   * Some ROMs have shadow registers

10. DESIGN COST

   * Use standard estimating procedures based on number of chips

   * Must add Engineering time to create Test Patterns and GMRs

   * May need debug time to hardware verify proper operation

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: LONG TUTORIAL	**PAGE** 6 of 14

SUBCATEGORY: BIT TECHNIQUE ATTRIBUTES

DATA TYPE: TEXT ☐   LIST ☒   TABLE ☐   GRAPHIC ☐   EQUATIONS ☐

DATA:

ON-BOARD BIT ROM
ATTRIBUTES
(CONT)

11. SOFTWARE DESIGN COST

    * Only applicable at system level

12. NUMBER OF BYTES OF STORAGE REQUIRED

    * Function of complexity of circuit (see Real Estate Penalty)

13. STAND-ALONE (SELF-CONTAINED BIT)?

    * Yes

14. WEIGHT

    * Proportional to real estate penalty weight

    * PENALTY = (Percent Real Estate Penalty) x (Weight of circuit)

15. Commercially available integrated circuits with testability features ROMs are available with shadow registers.

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: LONG TUTORIAL

SUBCATEGORY: DEFAULT DESIGN

DATA TYPE: TEXT ☐   LIST ☒   TABLE ☐   GRAPHIC ☐   EQUATIONS ☐

DATA:

a) See figure 3 for ON-BOARD ROM LEVEL II BLOCK DIAGRAM.

b) See figure 4 for TEST PATTERN AND GOOD MACHINE RESPONSE ROM DEFAULT DESIGN.

c) See figure 5 for GOOD MACHINE RESPONSE COMPARISON LOGIC DEFAULT DESIGN.

d) See figure 6 for INPUT MULTIPLEXER DEFAULT DESIGN.

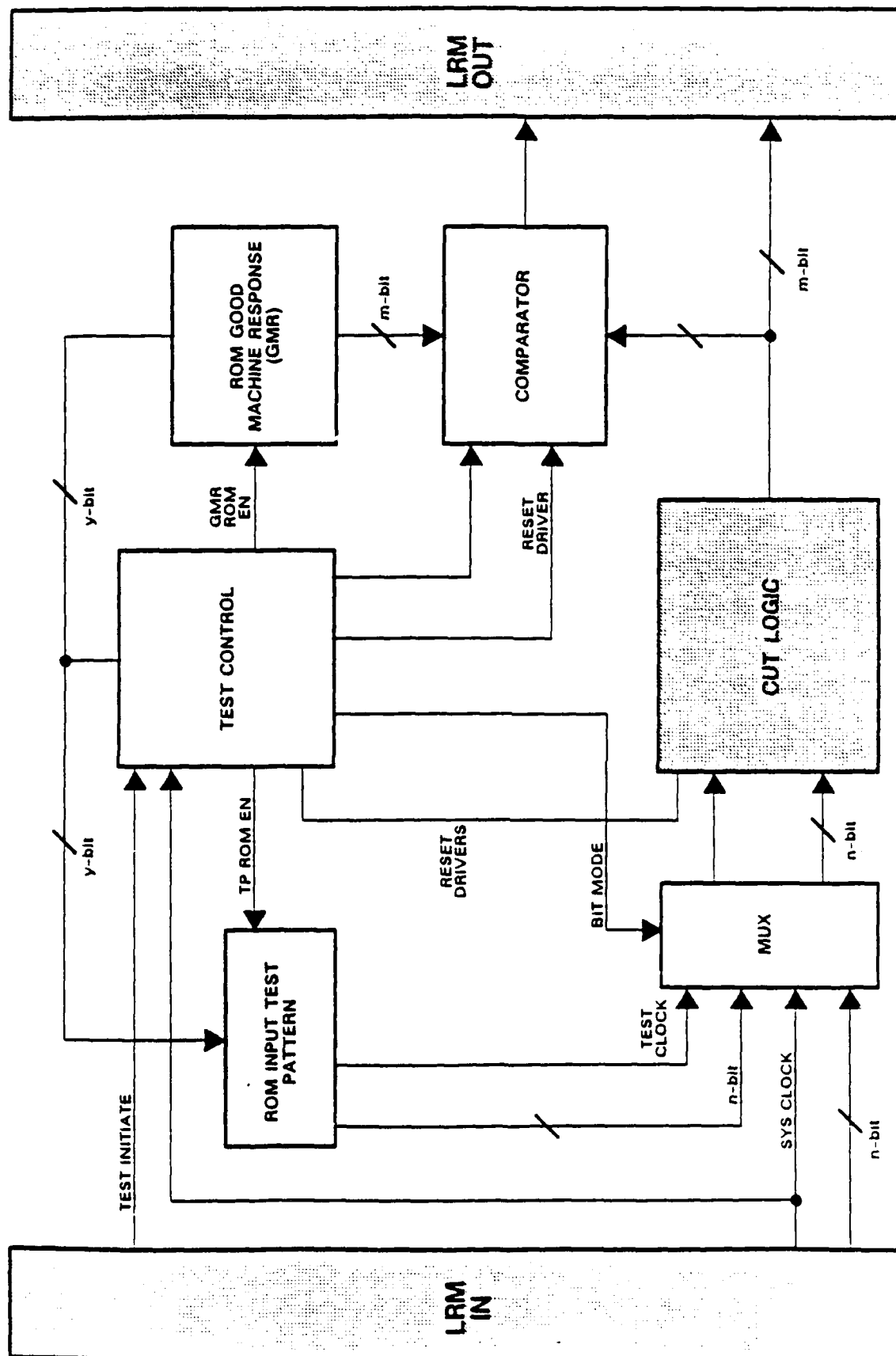e) See figure 7 for CONTROL LOGIC FOR ON-BOARD ROM DEFAULT DESIGN.
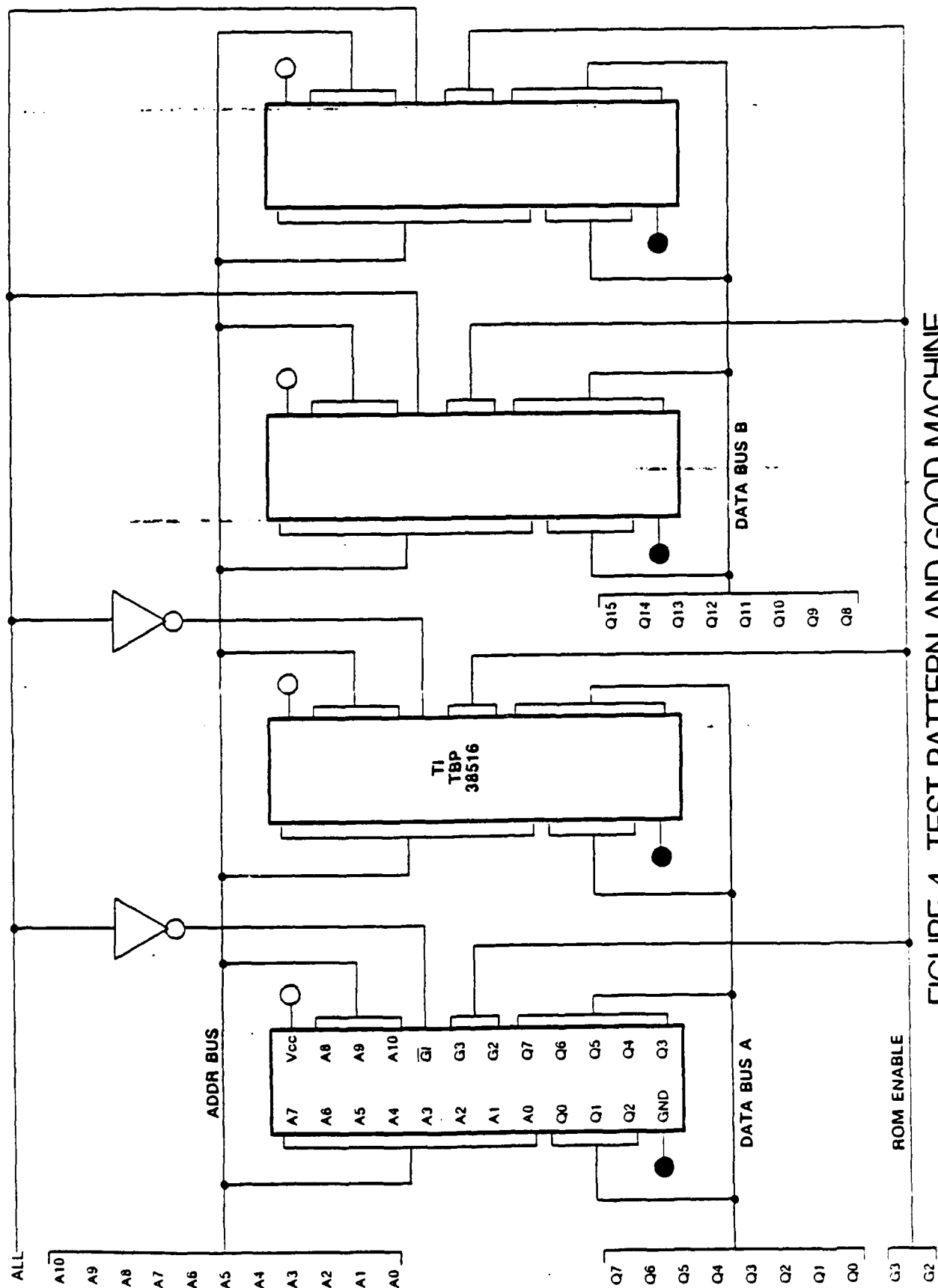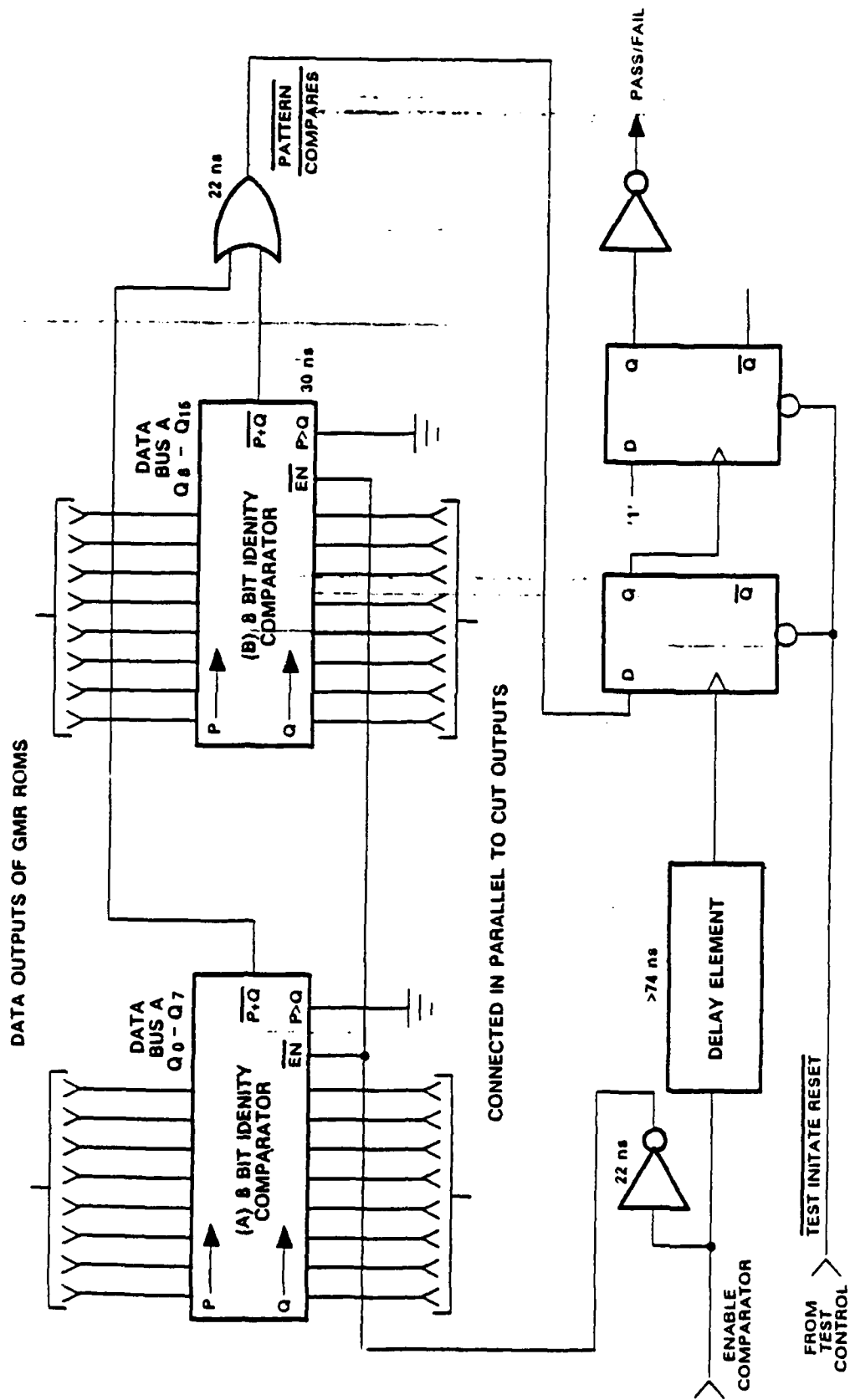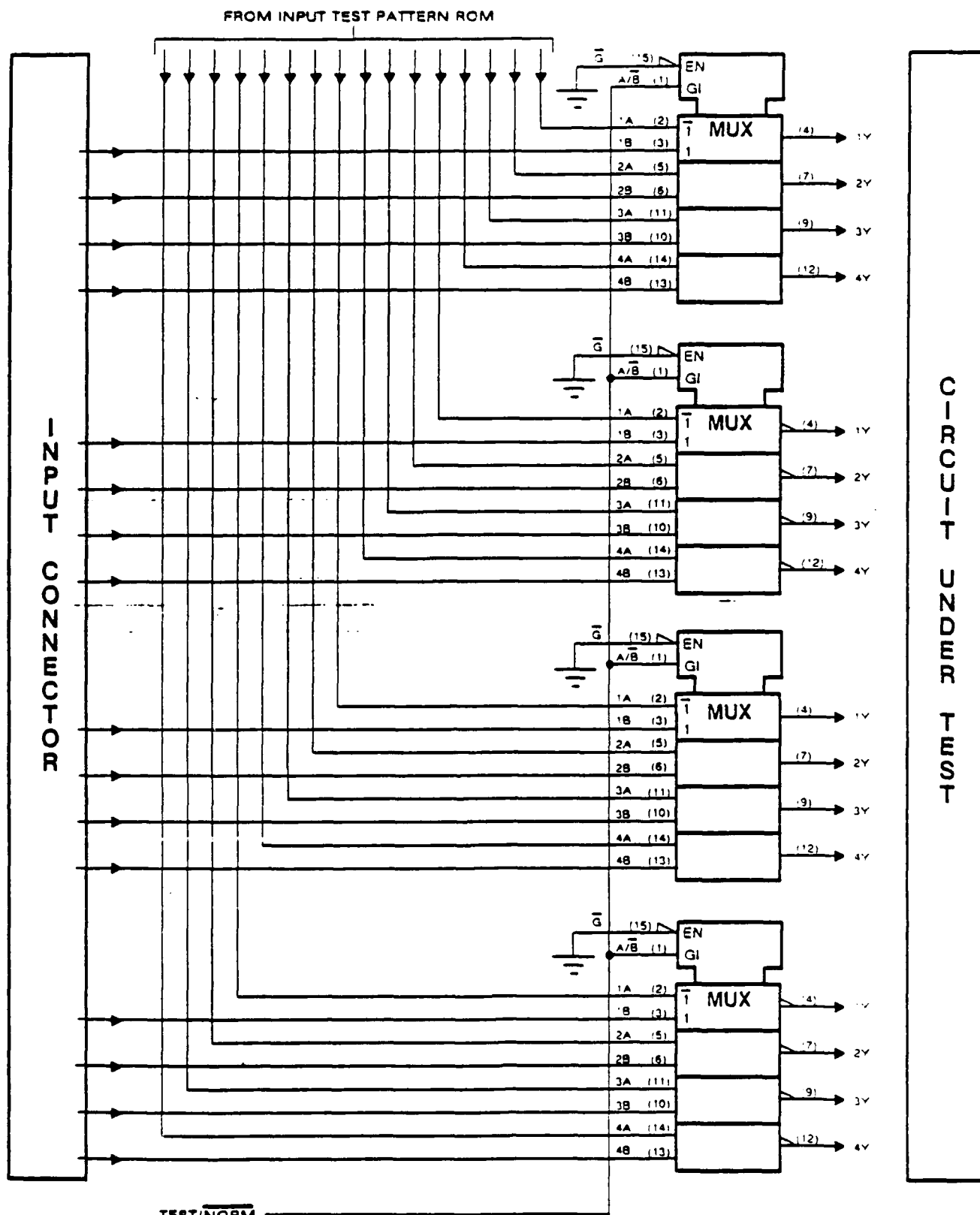
FIGURE 3 ON-BOARD ROM LEVEL II BLOCK DIAGRAM

A-12

FIGURE 4 TEST PATTERN AND GOOD MACHINE
RESPONSE ROM

DATA OUTPUTS OF GMR ROMS

22 ns

PATTERN
COMPARES

PASS/FAIL

DATA
BUS A
$Q_8 - Q_{15}$

$\overline{P+Q}$

30 ns

(B) 8 BIT IDENITY
COMPARATOR

$\overline{EN}$ P>Q

P

Q

DATA
BUS A
$Q_0 - Q_7$

$\overline{P+Q}$

(A) 8 IDENITY
COMPARATOR

$\overline{EN}$ P>Q

P

Q

CONNECTED IN PARALLEL TO CUT OUTPUTS

D   Q

'1'

$\overline{Q}$

D   Q

$\overline{Q}$

>74 ns

DELAY ELEMENT

22 ns

ENABLE
COMPARATOR

FROM
TEST
CONTROL

$\overline{TEST\ INITATE\ RESET}$

ns – nanosecond delay

FIGURE 5 GOOD MACHINE RESPONSE COMPARISON LOGIC

A–14

FROM INPUT TEST PATTERN ROM

NOTE: 2 QUAD 2 TO 1 LINE DATA SELECTOR/MUX's CAN BE REPLACED BY AN OCTAL 2 INPUT
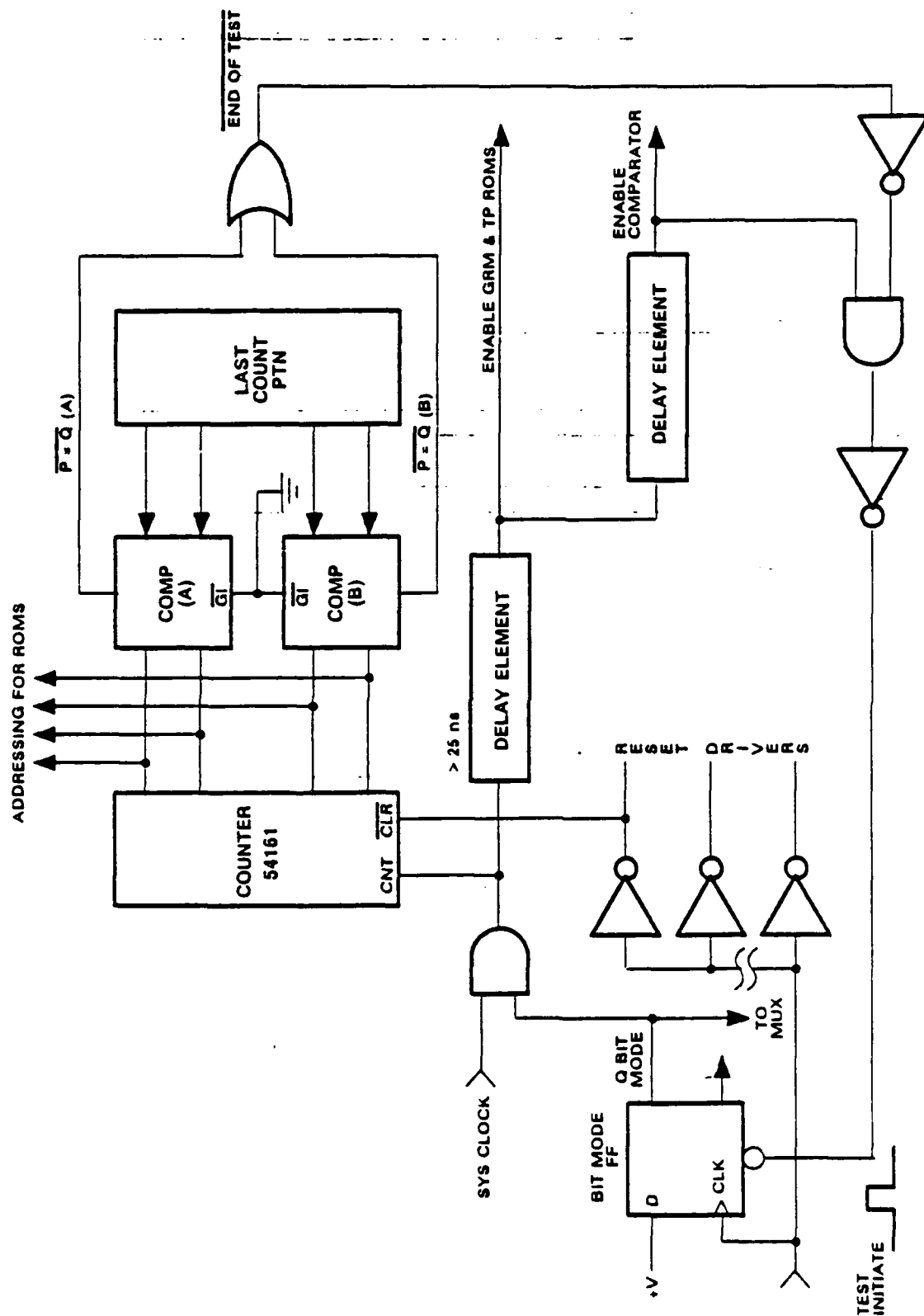MUXED LATCH–LS604.

FIGURE 6 INPUT MULTIPLEXER

PAGE 11 of 14

A–15

FIGURE 7 CONTROL LOGIC FOR ON-BOARD ROM

ns - nanosecond delay

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: LONG TUTORIAL

SUBCATEGORY: PART DATA TABLE

DATA TYPE: TEXT ☐   LIST ☐   TABLE ☒   GRAPHIC ☐   EQUATIONS ☐

DATA:

ON-BOARD ROM
PART DATA TABLE

NUMBER/NAME	AREA (sq in)	# OF PINS	POWER TYPICAL(mW)	POWER MAX. (mW)	WEIGHT (gms)
TBP385L16 2K x 8 PROM	0.375	24	325	500	6.5
74LS604/ OCT 2-IN MUXs LATCHES	0.87	28	275	350	7.5
74LS686/ 8 BIT MAG/ IDENT COMP	0.375	24	220	375	6.5
741617/ 4 BIT SYNC BIN COUNTER	0.243	16	315	455	2
7404/ HEX INVERTERS	0.243	14	90	165	2
7400/ QUAD 2-IN POS NAND	0.243	14	60	110	2
74125/ QUAD D FLIP FLOP	0.243	16	55	90	2

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: LONG TUTORIAL

SUBCATEGORY: BIBILOGRAPHY

DATA TYPE: TEXT ☐   LIST ☒   TABLE ☐   GRAPHIC ☐   EQUATIONS ☐

DATA:

NONE REQUIRED

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: USER REQUESTED DATA	PAGE 1 of 1

SUBCATEGORY:

DATA TYPE: TEXT ☐   LIST ☒   TABLE ☐   GRAPHIC ☐   EQUATIONS ☐

DATA:

	QUESTIONS	VARIABLE ASSIGNMENT
1.	How many primary input pins are used by the PCB's operational circuitry?	v1
2.	How many primary output pins are used by the PCB's operational circuitry?	v2
3.	How many test patterns are required to be stored in the ROMs?	v3
4.	What is the test pattern application rate?	v4
5.	What is the estimated initialization time?	v5

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: EQUATIONS

SUBCATEGORY: (DATA NOT TO BE DISPLAYED)

DATA TYPE: TEXT ☐ LIST ☐ TABLE ☐ GRAPHIC ☐ EQUATIONS ☒

DATA:

I) VARIABLE DEFINITIONS

n1= Number of ROM chips

n2= Number of MUX chips

n3= Number of COMPARATOR chips

n4= Number of COUNTER chips

n5= Number DECODE chips

n6= Number of PROGRAMMABLE DELAY chips

n7= BIT MODE status FF

n8= Number of CONTROL GATES

v1= Number of INPUT PINS<= 120

v2= Number of OUTPUT PINS<= 120

v3= Number of TEST PATTERNS<= 12288

v4= PATTERN RATE

v5= INITIALIZATION TIME

II) COMPONENT DETERMINATION EQUATIONS

$n1 = (v1/8)*(v3/2048) + (v2/8)*(v3/2048)$

$n2 = (v1/8)$

$n3 = (v2/8)$

$n4 = (v3/16)$

$n5 = $ Integer of $((n4 + 1)/2)$

$n6 = 2$

$n7 = 1$

$n8 = 2$

LIBRARY ELEMENT DATA SHEET

BIT TECHNIQUE: ON-BOARD ROM

CATEGORY: EQUATIONS	PAGE 2 of 2

SUBCATEGORY: (DATA NOT TO BE DISPLAYED)

DATA TYPE: TEXT ☐   LIST ☐   TABLE ☐   GRAPHIC ☐   EQUATIONS ☒

DATA:

III) PENALTY EQUATIONS

   a) AREA (sq in)

$$\text{Area of BIT chips} = (.375)n_1 + (.87)n_2 + (.375)n_3 + (.243)n_4 + (.375)n_5 + (.375)n_6 + (.243)n_7 + (.243)n_8$$

$$\text{Total area of BIT circuitry} = (\text{Area of BIT chips}) + 15\% \text{ for PC traces} = 1.15 \, (\text{ Area of BIT chips})$$

   b) POWER (mW)

$$\text{Power} = (325)n_1 + (350)n_2 + (375)n_3 + (455)n_4 + (375)n_5 + (200)n_6 + (90)n_7 + (110)n_8$$

   c) WEIGHT (gms)

$$\text{Weight of BIT chips (grams)} = (6.5)n_1 + (7.5)n_2 + (6.5)n_3 + (2.0)n_4 + (6.5)n_5 + (6.0)n_6 + (2)n_7 + (2)n_8$$

$$\text{Weight of BIT circuitry} = \text{Weight of BIT chips} + 10\% \text{ For Weight of solder} = 1.1 \,(\text{Weight of chips})$$

   d) TIME (ns)

$$\text{Test time} = (v_3)\,(v_4) + v_5$$

$$\text{Throughput delay} = 30$$

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C³I) activities. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C³I systems. The areas of technical competence include communications, command and control, battle management information processing, surveillance sensors, intelligence data collection and handling, solid state sciences, electromagnetics, and propagation, and electronic reliability/maintainability and compatibility.